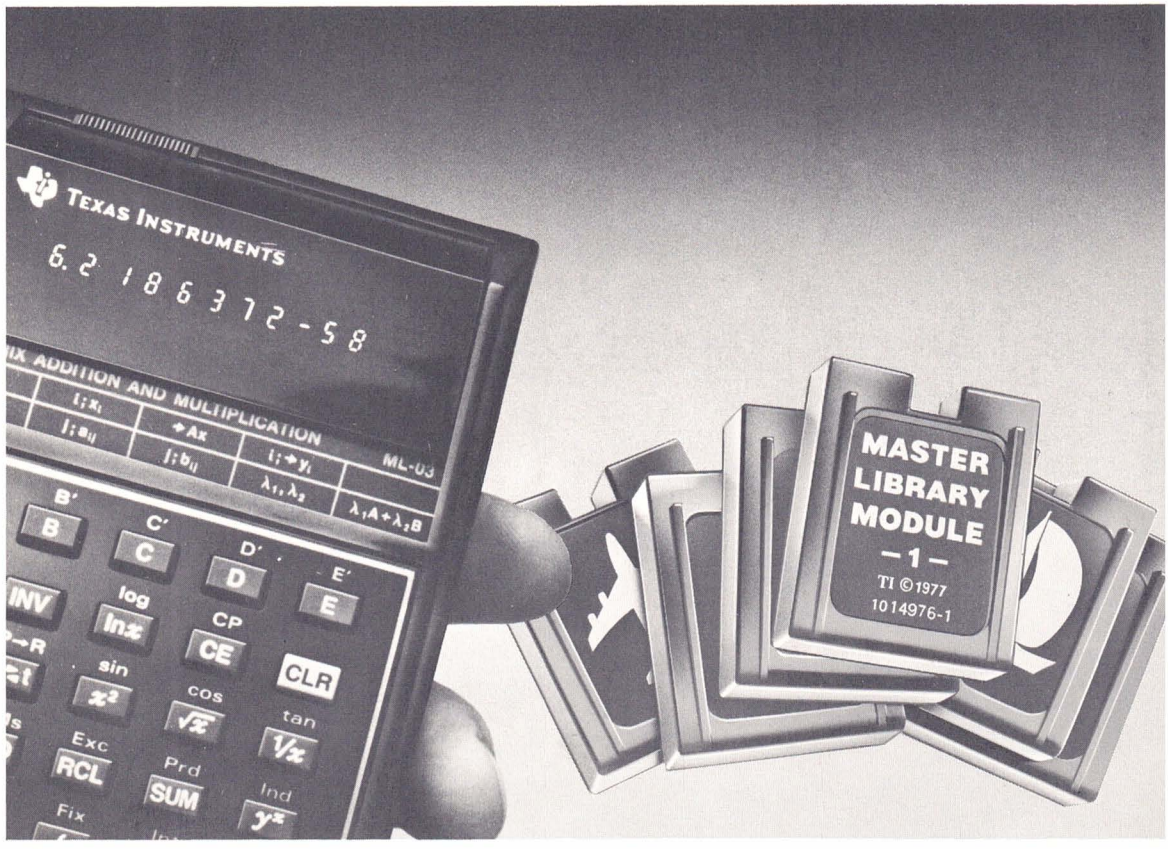
# Taschenrechner + Mikrocomputer

Anwendungsbereiche
Produktübersichten
Programmierung
Entwicklungstendenzen
Tabellen · Adressen

JAHRBUCH

1980 · · · ·

VIEWEG




# TI-58/59 + „Solid State Software™"- Ihre individuelle Problemlösung

Eine derartige Fülle von Möglichkeiten hat es bei Taschenrechnern noch nicht gegeben: Texas Instruments hat der modernsten Mikroprozessor-Technologie den Weg geebnet. Sie brauchen nur noch zuzugreifen: TI-58 mit 480 Programmschritten, TI-59 mit 960 plus Magnetkarten zur Speicherung Ihrer eigenen Programme. Beide Rechner verfügen zusätzlich über „Solid State Software™"-Steckmodule mit weiteren 5.000 Programmschritten.

Jeder Rechner wird mit einem Grundmodul geliefert mit Programmen für Mathematik, Umrechnung, Finanzwesen und allgemeine Anwendungen. Darüberhinaus gibt es eine ständig wachsende Anzahl von Steckmodulen, darunter Statistik, Baustatik, Vermessung, Finanzwesen (3 Stück) und das unterhaltsame Spiele-Modul. Programm-Handbücher und Programme von Spezialisten runden das Angebot ab.

Die einmalige Technologie erlaubt das Einprogrammieren individueller Lösungen, sofern der Modulbedarf eine gewisse Mindeststückzahl erreicht. Immer mehr Firmen machen davon Gebrauch.

Der dialogfähige Tisch-Thermodrucker PC-100 A/B/C garantiert dokumentenechte Protokollführung und spart Arbeitszeit. Er zeichnet auch Kurvenpunkte und druckt Texte aus.

Auch Ihre Probleme löst die ständig wachsende Software von Texas Instruments. Erkundigen Sie sich bei Ihrem Händler.

**TEXAS INSTRUMENTS**

Deutschland GmbH
Haggertystraße 1 · 8050 Freising

# Taschenrechner + Mikrocomputer

Jahrbuch 1980

## Herausgeber

Dr.-Ing. **Harald Schumny**
Bolchentwete 3
3300 Braunschweig

Wissenschaftlicher Mitarbeiter in der Physikalisch-Technischen Bundesanstalt in Braunschweig

## Autoren

Dr.-Ing. **Helmut Alt**
Mitarbeiter der Rheinisch-Westfälisches Elektrizitätswerk AG (RWE), Betriebsverwaltung Düren

**Paul Jürgen Dickers**
Mitarbeiter der Physikalisch-Technischen Bundesanstalt in Braunschweig

**Heiner Ensel**
Koblenz

Dr. **Helmut von Falser**
Mitarbeiter der Texas Instruments Deutschland GmbH

Dr.-Ing. **Christoph Fuchs**
Mitarbeiter der Hewlett-Packard GmbH

Ing. (grad.) **Karl-Heinz Gosmann**
Mitarbeiter der Hewlett-Packard GmbH

Ing. (grad.) **Max Gottlob**
Mitarbeiter der Texas Instruments Deutschland GmbH

Obering. **Wilhelm-Rüdiger Haberditz**
Mitarbeiter der Hartmann & Braun AG, Frankfurt

Dipl.-Ing. **Werner Handke**
Mitarbeiter bei Rohde & Schwarz in Köln

Dr. **Martin Hund**
Mitarbeiter der Leybold-Heraeus GmbH

Dipl.-Phys. **Rolf Krüger**
Wissenschaftlicher Mitarbeiter in der Physikalisch-Technischen Bundesanstalt in Braunschweig

**Volkhart Lehmann**
Studienrat in Koblenz

Dipl.-Ing. **Wolfgang Link**
Oberstudienrat an der Fachschule für EDV in Paderborn

Dipl.-Volkswirt **Roland Löhr**
Herausgeber der Fachzeitschrift „65XX MICRO MAG"

Ing. (grad.) **Harald Nahrstedt**
Dozent an der VHS Hamm

Dr. **Dieter Nenner**
Mitarbeiter der Texas Instruments Deutschland GmbH

Prof. Dr. **Otger Neufang**
Dozent an der FH Düsseldorf

Dr.-Ing. **Wolfgang Schneider**
Dozent an der FH Wolfenbüttel

Dr.-Ing. **Harald Schumny**
Wissenschaftlicher Mitarbeiter in der Physikalisch-Technischen Bundesanstalt in Braunschweig

Ing. (grad.) **Bernd Strößner**
Geschäftsführer der Micronic Computer Systeme (MCS) in Berlin

Dr. **Hans-Joachim Stuckenberg**
Mitarbeiter am DESY (Deutsches Elektronen-Synchrotron) in Hamburg

Dipl.-Ing. **Peter Welzel**
Studiendirektor an der Fachschule für EDV in Paderborn

Prof. Dr. **Alexander Wynands**
Professor für Didaktik der Mathematik an der Pädagogischen Hochschule Rheinland – Abteilung Bonn

# Taschenrechner + Mikrocomputer

# Jahrbuch 1980

Anwendungsbereiche
Produktübersichten
Programmierung
Entwicklungstendenzen

Mit zahlreichen Bildern, Tabellen und Adressen

Herausgegeben von Harald Schumny

Friedr. Vieweg & Sohn Braunschweig/Wiesbaden

CIP-Kurztitelaufnahme der Deutschen Bibliothek

**Taschenrechner- + [und] Mikrocomputer-Jahrbuch:**
Anwendungsbereiche, Produktübersichten, Programmierung, Entwicklungstendenzen; mit zahlr. Bildern, Tab. u. Adressen. – Braunschweig, Wiesbaden: Vieweg.
Erscheint jährl.

1980 ff. – 1979 ff.

## Verzeichnis der Inserenten


| | Seite |
|---|---|
| Texas Instruments | U 2 + U 4 |
| elrad – Magazin für Elektronik | 52 |
| Leybold-Heraeus | 101 |
| elrad – Triton Computer | 129 |
| MCS – Micronic Computer Systeme | 148 + 256 |
| Commodore | 152 |
| MICRO MAG | 236 |
| Chip | 257 |
| Hewlett Packard | U 3 |


Alle Rechte vorbehalten
© Friedr. Vieweg & Sohn Verlagsgesellschaft mbH, Braunschweig 1979

Die Vervielfältigung und Übertragung einzelner Textabschnitte, Zeichnungen oder Bilder, auch für Zwecke der Unterrichtsgestaltung, gestattet das Urheberrecht nur, wenn sie mit dem Verlag vorher vereinbart wurden. Im Einzelfall muß über die Zahlung einer Gebühr für die Nutzung fremden geistigen Eigentums entschieden werden. Das gilt für die Vervielfältigung durch alle Verfahren einschließlich Speicherung und jede Übertragung auf Papier, Transparente, Filme, Bänder, Platten und andere Medien.

Satz: Friedr. Vieweg & Sohn, Braunschweig / Wiesbaden
Druck: C. W. Niemeyer, Hameln
Buchbinder: W. Langelüddecke, Braunschweig
Printed in Germany

ISBN 3-528-04154-4 ISSN 0172-4053

# Vorwort

Verehrter Leser,

etwa zehn Jahre sind vergangen, seit die ersten Mikroprozessoren und Taschenrechner auf den Markt kamen. Seit etwa acht Jahren werden diese „Kleinstcomputer" mit den vorher kaum für möglich gehaltenen Fähigkeiten in großen Stückzahlen benutzt. Und nur rund fünf Jahre ist es her, seit programmierbare Taschenrechner und „Jedermann-Computer" *(Personal Computer)* zur Verfügung stehen.

Begleitet wurde diese rasante technische Entwicklung von einer regen publizistischen Tätigkeit, die sich allerdings zumindest in den ersten Jahren überwiegend auf die USA beschränkte. Produktbeschreibungen, Fachartikel und Bücher lagen mithin zunächst fast immer nur in englischer Sprache vor. Nun sind viele der wichtigsten Arbeiten übersetzt. Aber auch deutsche Autoren und Verlage sorgen in zunehmendem Maße für fachlichen Lesestoff. Das hat wesentlich zum inzwischen relativ hohen Spezialwissen bei den vielen Benutzern beigetragen. Jedoch ist eine andere Begleiterscheinung die wachsende Schwierigkeit, aus der Fülle der Veröffentlichungen das jeweils geeignetste herauszufinden.

Herausgeber und Verlag haben sich aus diesem Grund entschlossen, das „Taschenrechner + Mikrocomputer Jahrbuch" zu entwickeln. Es soll damit nämlich einerseits eine systematische und aktuelle Fachinformation entstehen, die einen schnellen und gezielten Zugriff gestattet. Andererseits werden mit vergleichenden Übersichten, technischen Daten, Adressen usw. zuverlässige Orientierungshilfen geboten.

Das Jahrbuch-Konzept geht von einer Zweiteilung aus: Der *Fachteil* enthält aktuelle Beiträge zu den Bereichen *Taschenrechner, Mikrocomputer* und *Speicher.* Der *Datenteil* besteht im wesentlichen aus Produktübersichten, Bezugsquellen, Adressen, Literatur- und Zeitschriftenlisten. Ein detailliertes *Sachwortverzeichnis* gestattet das Auffinden spezieller Textstellen.

Um auch weniger erfahrenen Lesern entgegenzukommen, haben die Fachbeiträge ein Glossar. Darin sind in knapper Form Fachausdrücke erklärt, die im jeweiligen Aufsatz ohne Kommentar benutzt werden. Diese Fachausdrücke sind im Text und im Glossar (in der Regel auf derselben Seite) *halbfett* gesetzt.

In der vorliegenden ersten Ausgabe „Taschenrechner + Mikrocomputer Jahrbuch 1980" sind die Produktübersichten (Datentabellen) so vollständig wie irgend möglich gehalten. Das bedeutet, es sind auch Taschenrechner, Tischrechner, Mikroprozessoren sowie Mikrocomputer aufgenommen, die nicht mehr hergestellt und verkauft werden. Dadurch entsteht aber gerade der beabsichtigte systematische Überblick. Und es werden beispielsweise Besitzer älterer Geräte in die Lage versetzt, eine Entscheidung darüber zu treffen, ob eine Neuanschaffung lohnt oder gar notwendig ist.

Die nachfolgenden Ausgaben des Jahrbuches werden im Datenteil nur noch aktuelle Informationen enthalten und vor allem Neuentwicklungen und Trends herausstellen. So ist dafür gesorgt, daß eine kontinuierliche Fortschreibung der raschen Entwicklung entsteht, die den hoch- und höchstintegrierten elektronischen Bauteilen und den daraus gefertigten Geräten nun einmal anhaftet.

Das gewählte Konzept stellt sicher, daß ältere Jahrbücher nicht von vornherein zum Bücherschrank- oder Papierkorbfüller abgewertet sind. Vielmehr werden sie als oft genutzte Nachschlagewerke und als Chronik der „Mikrorechnerentwicklung" über Jahre hin zur Verfügung stehen.

Herausgeber und Verlag

Braunschweig, Wiesbaden im August 1979

# Inhaltsverzeichnis

Leitartikel: Vom Abakus zum Mikroprozessor . . . . . 1

## Taschenrechner 3

*H. Alt*
Historischer Überblick . . . . . 5

*V. Lehmann / H. Ensel*
Korrektur von Multiple-Choice-Tests mit dem TI-59 . . 7

*W. R. Haberditz*
Anwendung des programmierbaren Taschenrechners TI-59 bei der Temperaturbestimmung an elektrischen Wicklungen . . . . . 15

*W. R. Haberditz*
Basis-Umwandlung von Zahlen mit dem TI-59 . . . . . 21

*H. Alt*
Tabellenfreie Berechnung der Einkommensteuer mit Hilfe des programmierbaren Taschenrechners . . . . . 25

*Ch. Fuchs / K.-H. Gosmann*
UPN – Die zukunftsorientierte Computer-Logik . . . . . 28

*A. Wynands*
Taschenrechner im Unterricht . . . . . 32

*D. Nenner*
Intelligente Taschenrechner mit AOS – Auswahlkriterien für Schulrechner . . . . . 44

*H. Nahrstedt*
Programmierbare Taschenrechner und Mikrocomputer in der Erwachsenenbildung . . . . . 48

## Mikrocomputer 53

*H. Schumny*
Mikrocomputer – Grundbegriffe und Abgrenzungen . . 55

*H. v. Falser*
Planen–Lernen–Gestalten; Das Programm des individuellen Dialogrechners . . . . . 61

*H. Schumny*
Personal Computer . . . . . 67

*W. Handke*
Mikroprozessoren – Hilfsmittel zur Programmentwicklung und Testmethoden . . . . . 69

*H. Schumny*
Schnittstellen und Busanordnungen . . . . . 79

*P. Welzel*
Die Adressierung in Mikrocomputer-Systemen . . . . . 86

*P. Welzel*
Interrupts in Mikroprozessor-Systemen . . . . . 95

*W. Schneider*
Grafische Darstellung von Programmabläufen . . . . . 102

*W. Schneider*
Dialogsprache BASIC . . . . . 108

*M. Hund*
Grundausbildung Mikroprozessor und Mikrocomputer – Eine experimentelle Einführung in Struktur und Arbeitsweise . . . . . 121

*M. Gottlob*
Das TI-Lern- und -Lehrsystem TM990/189 . . . . . 126

*R. Löhr*
Impulserzeugung mit dem R6522 VIA (Versatile Interface Adapter) . . . . . 130

*B. Strößner*
ALPHA 1 und BETA 8 – Ein Mikrocomputersystem aus Berlin . . . . . 134

*H.-J. Stuckenberg*
Bipolare, schnelle Mikroprozessorsysteme . . . . . 138

*R. Krüger / P. J. Dickers*
Schnelle Mathematik für den TMS 9900 . . . . . 146

*R. Krüger / P. J. Dickers*
Mathematik-Assembler für Mikroprozessoren . . . . . 149

## Speicher 153

*H. Schumny*
Aufzeichnungsverfahren für magnetische Datenträger . . 155

*H. Schumny*
Beispiele für die Übertragung, Speicherung und Sicherung von Daten . . . . . 160

*O. Neufang*
Halbleiterspeicher . . . . . 166

*W. Link*
Programmierbare Festwertspeicher (PROM) . . . . . 171

*H. Schumny*
Bipolare Speicher-Technologien . . . . . 177

*W. Link*
CCD-Halbleiterspeicher ersetzen rotierende Massenspeicher . . . . . 181

## Daten 185

Produktübersichten . . . . . 186

Adressen . . . . . 237

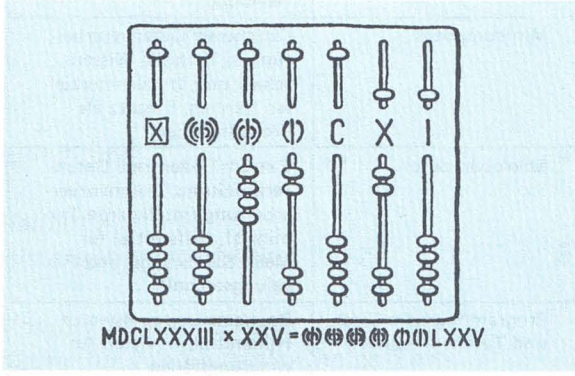
Bücher, Zeitschriften . . . . . 246

## Sachwortverzeichnis . . . . . 251

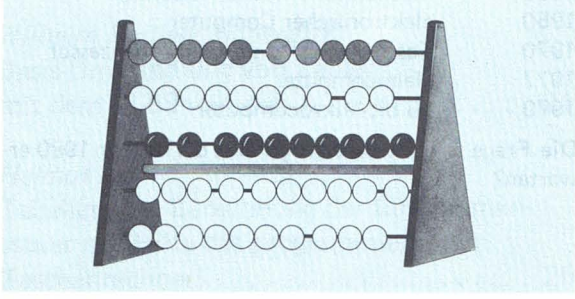
H. Schumny

# Vom Abakus zum Mikroprozessor

Abakus hieß der „Tischrechner" der Antike. Es handelte sich um ein Rechenbrett mit einer Reihe von Querrinnen, in denen Kügelchen oder Steinchen hin- und hergeschoben wurden, die verschieden groß, verschieden geformt oder gefärbt waren. Durch Gruppenbildungen und geschickte Wertezuordnungen konnten damit alle in der antiken Geschäftswelt und im Privatbereich nötigen Berechnungen ausgeführt werden. Auch sehr reiche Römer kamen in der Regel mit dem möglichen Wertebereich und der Stellenzahl aus: Zahlen bis 10 Millionen konnten bestimmt verarbeitet werden.




Eine Abakus-Version, bei der verschiedenfarbige Kugeln auf übereinander angeordneten Stäben verschoben werden können, ist sicher vielen von uns schon einmal begegnet. In Rußland beispielsweise soll dieses Rechengerät manchmal noch heute als „Kassenterminal" dienen.



Es darf aber nicht übersehen werden, daß der Abakus keine *Rechenmaschine* ist. Es handelt sich lediglich um eine manuell (von Hand also) zu bedienende Rechenhilfe, bei der praktisch nur das Abzählen an Fingern oder mit Steinchen in eine systematische Form gebracht worden ist. Dieses systematische Abzählen finden wir auch beim römischen „Zahlensystem" wieder, bei dem ein Strich I für eine Einheit steht. Fünf, zehn, 50 oder 100 etc. Einheiten werden zusammengefaßt und mit einprägsamen Symbolen V, X, L oder C etc. dargestellt.

Die Entwicklung von *Rechenmaschinen* begann erst gegen Ende des für bildende Künste, Baukunst und Wissenschaften so bedeutenden Zeitraumes, der heute *Renaissance* genannt wird. 1632, also während des Dreißigjährigen Krieges, baute *Wilhelm Schickard* in Tübingen seine „Rechenuhr". *Helmut Alt* gibt in seinem Aufsatz auf Seite 5 dieses Jahrbuches einen „Historischen Überblick", der von dieser ersten Maschine bis zum modernen Taschenrechner reicht. Im Zeitdiagramm auf der nächsten Seite sind die wichtigsten Entwicklungsschritte in den vergangenen 2000 Jahren eingetragen, wobei für die Zeitachse eine exponentielle Darstellung gewählt wurde.

Sieht man von der durch *Hollerith* etwa 1880 begonnenen Entwicklung einer Lochkartenmaschine ab, vergingen seit *Schickard* mehr als 300 Jahre, bis *Aiken* 1937 und *Zuse* 1941 ihre ersten elektromechanischen Computer bauten, bei denen Relais als Schaltelemente eingesetzt wurden. 1946 wurde von *Eckert* und *Mauchly* der erste Röhren-Computer ENIAC fertiggestellt. Erst nach 1950 wurden erstmals Transistoren eingebaut. Ab etwa 1960 folgte die „dritte Computer-Generation" mit integrierten Schaltkreisen (IC). Um 1965 entstanden die ersten *Minicomputer* (PDP-8). Ergebnisse der seither ansteigenden Integrationsdichte bei integrierten Schaltkreisen waren die von der Firma INTEL ab 1969 entwickelten Bausteine 4004 und 8008 – der *Mikroprozessor* war geboren. Etwa gleichzeitig entstanden die ersten Taschenrechner.

Man sollte an dieser Stelle einmal nachdenken: Nicht einmal zehn Jahre sind vergangen, seit elektronische

Taschenrechner und Mikroprozessoren vorgestellt wurden. Jedes Jahr in dieser Dekade brachte neue Überraschungen. Und man hat sich fast schon daran gewöhnt, daß die Integrationsdichte und die Fähigkeiten der „Mikrorechner" weiter steigen, gleichzeitig aber die Preise heruntergehen.

Drei Höhepunkte haben die letzten Jahre geprägt:

1. Die Aufwertung der programmierbaren Taschenrechner durch Magnetstreifen, Drucker, nichtflüchtige Speicher und Steckmodule, die in einem Beispiel bis zu 5000 Programmschritte speichern können;
2. Die Vorstellung des PET 2001, des ersten vollständigen und preisgünstigen Heimcomputers (andere Bezeichnungen: *Hobby-Computer, Home Computer, Personal Computer* oder auch *Mehrzweck-Computer* bzw. *General Purpose Computer,* GPC);
3. Die Auslieferung hochentwickelter 16 bit-Mikroprozessoren.

Die mit Mikroprozessoren der letzten Generation entwickelten Mikrocomputer sind zumindest teilweise leistungsfähiger als etwa 1970 gebaute Minicomputer. Bei diesem Vergleich können dann „nur noch" Preise und Gehäuseabmessungen mit „Mikro" beschrieben werden. Der in BASIC programmierbare *Taschen-Computer* in den Abmessungen eines Taschenrechners ist sicher kein Hirngespinst mehr. Noch werden aber in sehr großer Zahl die verschiedenen in den letzten zehn Jahren gefertigten Rechner und Computer genutzt. In der nachfolgenden Aufstellung ist darum eine Einteilung in *Rechnerklassen* vorgenommen, die diesem technischen Status quo Rechnung trägt.

| Rechnerklasse | Anwendungsbereiche |
|---|---|
| Großcomputer | Spezialrechner für Technik, Wissenschaft und Rechenzentren |
| Mittlere Anlagen | Kommerzieller Bereiche (Mittlere Datentechnik, MDT) |
| Prozeßrechner | Meßdatenverarbeitung, Überwachung, Steuerung, Regelung |
| Minicomputer | Dezentrale Datenverarbeitung in Technik, Wissenschaft und im kommerziellen Bereich, Einsatz als Prozeßrechner |
| Mikrocomputer | Extrem Dezentrale Datenverarbeitung, Datenvorverarbeitung (intelligente Terminals), Hilfsmittel für Meß-, Steuerungs- und Regelungstechnik |
| Programmierbare Tisch- und Taschenrechner | Bequeme und preiswerte Hilfsmittel in vielen Anwendungsfällen |

Möglicherweise wird schon im nächsten Jahrbuch eine Revision dieser Klassifizierung nötig sein. Vorerst festhalten können wir folgende Entwicklungsschritte:

| | |
|---|---|
| 100 v. Chr. | Abakus |
| 1600 | mechanische Rechenmaschine |
| 1950 | elektronischer Computer |
| 1970 | Taschenrechner und Mikroprozessor |
| 1977 | Heimcomputer |
| 1979 | 16 bit-Mikrocomputer |

Die Frage scheint berechtigt: Was dürfen wir 1980 erwarten?

# Taschenrechner

## Inhaltsübersicht


Einführung . . . . . 4

*Helmut Alt*
Historischer Überblick . . . . . 5

*Volkhard Lehmann* und *Heiner Ensel*
Korrektur von Multiple-Choice-Tests mit dem TI-59 . . . . . 7

*Wilhelm Rüdiger Haberditz*
Anwendung des programmierbaren Taschenrechners TI-59 bei der Temperaturbestimmung an elektrischen Wicklungen . 15

*Wilhelm Rüdiger Haberditz*
Basis-Umwandlung von Zahlen mit dem TI-59 . . . . . 21

*Helmut Alt*
Tabellenfreie Berechnung der Einkommensteuer mit Hilfe des programmierbaren Taschenrechners . . . . . 25

*Christoph Fuchs* und *Karl-Heinz Gosmann*
UPN – Die zukunftsorientierte Computer-Logik . . . . . 28

*Alexander Wynands*
Taschenrechner im Unterricht . . . . . 32

*Dieter Nenner*
Intelligente Taschenrechner mit AOS – Auswahlkriterien für Schulrechner . . . . . 44

*Harald Nahrstedt*
Programmierbare Taschenrechner und Mikrocomputer in der Erwachsenenbildung – Erfahrungen an der VHS Hamm 48

# Einführung

Was Taschenrechner sind, weiß natürlich jeder. Man bekommt sie ja fast geschenkt, wenn man ein Kilo Kaffee kauft oder jemanden für einen Lesering wirbt. Andererseits gibt es auch solche wertvollen Exemplare, daß sie in den Fachabteilungen der Kaufhäuser nur in verschlossenen Glaskästen ausgestellt werden. Wir sollten darum hier die Bezeichnung „Taschenrechner" doch etwas differenzierter betrachten. Aus der Benutzersicht kann dazu folgende Einteilung nützlich sein:

- Vierspeziesrechner
- Kaufmännische Rechner
- Technisch-wissenschaftliche Rechner
- Programmierbare Rechner.

*Vierspeziesrechner* heißen die billigen Massenartikel, weil sie im wesentlichen nur die vier Grundrechenarten beherrschen. Die Eingabe- und Verarbeitungsmethode ist in der Regel *algebraisch* (AL: *Algebraische Logik*). Das bedeutet, Aufgaben werden so eingetastet und gelöst, wie man sie spricht, also z.B. „3 plus 4 ist gleich".

*Kaufmännische Rechner* nennt man Geräte, die für die Lösung kaufmännischer Probleme spezialisiert sind, die also die Durchführung solcher Aufgaben wie Saldierung, Kontenführung, Steuer- und Zinsberechnungen etc. erleichtern. Kaufmännische Rechner sind oft nur einfach ausgestattet, zuweilen aber auch programmierbar. Gemeinsam ist dieser Rechnergruppe jedoch nahezu immer, daß eine Eingabemethode verwendet wird, die *arithmetisch* heißt (ARL: *Arithmetische Logik*). Kennzeichnend hierfür ist die Eingabe des Vorzeichens erst nach der Operandeneingabe.

*Technisch-wissenschaftliche Rechner* sind Geräte der gehobenen Preisklasse, die auf Tastendruck hin mathematische, technische, statistische usw. Funktionen lösen können. Neben der Eingabemethode AL sind hier zwei weitere von Bedeutung: AOS *(Algebraisches Operationssystem)* als Verbindung von AL mit Klammern und Hierarchie sowie UPN *(Umgekehrte Polnische Notation)*.

*Programmierbare Taschenrechner* schließlich bilden die höchste Klasse (von den Fähigkeiten und vom Preis her). Sie sind technisch-wissenschaftlich oder kaufmännisch ausgerichtet, die Eingabemethode heißt demzufolge entweder ARL, AL, AOS oder UPN. Die am höchsten entwickelten Vertreter dieser Klasse sind mit Magnetstreifenlesern ausgestattet oder erlauben das Einstecken von Halbleiterspeichern mit ganzen Programmbibliotheken.

Vorangestellt ist dem Fachteil „Taschenrechner" ein „historischer Überblick". Den Hauptteil bilden Aufsätze mit Beispielen zur Nutzung programmierbarer Taschenrechner, die von der „Multiple-Choice-Testkorrektur" bis zur „tabellenfreien Berechnung der Einkommensteuer" reichen. Eine letzte Gruppe besteht aus Beiträgen zur Verwendung von Taschenrechnern im Unterricht. Während einer sich nahezu ausschließlich auf den nicht programmierbaren, einfachen elektronischen Rechner beschränkt und z.B. die „fachdidaktische Komponente" untersucht, schlägt der letzte Beitrag „Programmierbare Taschenrechner und Mikrocomputer in der Erwachsenenbildung" sozusagen eine Brücke zum zweiten Fachteil „Mikrocomputer" dieses Jahrbuches.

*Behauptung:* Taschenrechner sind erst möglich, seit integrierte elektronische Bauteile raumsparend in einem Taschen-gerechten Gehäuse untergebracht werden können – also erst seit etwa 10 Jahren. *Falsch!* – Bereits 1666 gab es eine Taschenrechenmaschine. Der folgende historische Überblick beschreibt auch diesen „Meilenstein der Rechnergeschichte" und zeigt den Weg bis zum modernen programmierbaren Taschenrechner auf.

H. Alt

# Historischer Überblick

Die Anfänge der Entwicklung von Rechenmaschinen gehen zurück auf *Wilhelm Schickard* (1592–1635), Professor an der Universität Tübingen [1, 2]. Er baute im Jahre 1623 eine Maschine „Rechenuhr" für die vier Grundrechenarten, bei der für das Addier- und Subtrahierwerk erstmalig das dekadische Zählrad verwendet wurde. *Blaise Pascal* (1623–1662) vollendete seine erste Rechenmaschine für Addition und Subtraktion im Jahre 1645. [Die Maschine diente zur Erleichterung der täglichen Arbeit seines Vaters, der Steuerintendant war.] Ein Vorläufer unserer heutigen Taschenrechner war *Samuel Morland's* (1625–1695) Taschenrechenmaschine aus dem Jahre 1666. Diese war vom Pascalschen Typ, jedoch in den Abmessungen nur 3 x 4 x 1/4 Inch (76 x 102 x 6,3 mm) groß und für Multiplikationen geeignet. Das Original ist im britischen Science Museum in London zu sehen. Eine schon fortgeschrittene Rechenmaschine, bei der die Arbeitsgänge der Zahleneinstellung und der Rechnung voneinander getrennt waren und mit der man nicht nur Additionen und Subtraktionen, sondern auch Multiplikationen „mühelos" und Divisionen „fast mühelos" ausführen konnte, konstruierte *Gottfried Wilhelm Leibniz* (1646–1716) im Jahre 1672. Seine Motivation wird in einem Zitat aus dem Jahre 1671 besonders deutlich [3]:

"Indignum enim est excellentium virorum horas servili calculandi labore perire, quia Machina adhibita vilissimo cuique secure transcribi posset."

*Zu deutsch:*

„Eines geistig hochstehenden Mannes ist es unwürdig, seine Zeit zu vertun mit sklavischer Rechenarbeit, denn mit einer Maschine könnte auch der Allerdümmste die Rechnung sicher ausführen."

Nach nahezu 250-jähriger Pause in der Rechenmaschinenentwicklung konzentrierte man sich ab der Jahrhundertwende auf die Verbesserung der mechanischen Realisierung der **Vierspezies-Rechner.**

Im Sinne einer neuen Entwicklungsdimension gab *Konrad Zuse* (geb. 1910) im Jahre 1941 mit der ersten voll funktionstüchigen programmgesteuerten Rechenanlage der Welt, der Zuse Z3 in Relaistechnik, den Startschuß zu einer stürmischen Entwicklung auf dem Gebiet elektronischer Großrechenanlagen.

Diese Entwicklung war im wesentlichen gekennzeichnet durch die Fortschritte, die auf dem Sektor der Halbleitertechnologie erzielt wurden.

Mit der Verfügbarkeit hochintegrierter **monolithischer Schaltkreise** war der Verwirklichung leistungsfähiger wissenschaftlicher Rechner im Taschenformat die Tür geöffnet. Im Jahre 1970 wurden bereits 14 000 Schaltelemente auf einer Fläche von 17 Quadratmillimetern untergebracht und Zeiten für eine Rechenoperation im **Nanosekundenbereich** erzielt. Zu dieser Zeit erschienen auch die ersten wissenschaftlichen Taschenrechner auf dem Markt, freilich noch zu einem Preis in der Größenordnung des Monatseinkommens eines Ingenieurs. Die folgenden Jahre waren durch eine rasche Zunahme von Typenvielfalt, Leistungsfähigkeit und Preisverfall auf diesen Markt gekennzeichnet. Im Jahre 1974 erschienen die ersten programmierbaren Taschenrechner sowohl ohne als auch mit **externem Speicher.** Bei den Folgetypen wurde dann insbesondere der verfübare Speicherplatz und der Programmierkomfort weiter gesteigert.

Der Taschenrechner trat ab 1972 unaufhaltsam an die Stelle des bewährten Rechenschiebers, der im Jahre 1654 von *Robert Bissaker* eingeführt worden war. In der von *Seth Partridge* im Jahre 1657 verbesserten

Form, mit beweglicher Zunge zwischen den beiden an den Enden fixierten Körperteilen, wurde er in den folgenden 300 Jahren, nicht zuletzt wegen des ebenso mechanisch trivialen wie mathematisch genialen Aufbaus, zu einer höchst nützlichen „Rechenmaschine", die insbesondere im ingenieurwissenschaftlichen Bereich unverzichtbar geworden war.

Die Anwendung des Rechenschiebers im studentischen Bereich geht seit der offiziellen Zulassung der Taschenrechner als Hilfsmittel zu den Prüfungen an wissenschaftlichen Hochschulen und Fachhochschulen immer mehr zurück. Wegen der digitalen Anzeige elektronischer Rechner ist die Übersicht über einzelne Bereichsdekaden, wie man sie vom Rechenschieber gewöhnt war, nicht mehr gegeben. Die Abschätzung des Einflusses einzelner Faktoren auf das Rechenergebnis ist daher nicht mehr in anschaulicher Form möglich. Es bleibt zu hoffen, daß der hohe didaktische Nutzen, den viele Generationen aus der analogen Arbeitsweise mit dem Rechenschieber und dem Zwang zum *Mitdenken in Größenordnungen* zogen, bei der Anwendung des Taschenrechners auf andere Weise erreicht wird.

Inzwischen ist man an einem Punkt angelangt, wo die ausgereifte **Hardware** der programmierbaren Rechner eine intensive Investition in **Software**-Arbeiten rechtfertigt. Auf diesem Sektor ist praktisch ein unbegrenztes **Innovationspotential** vorhanden, aus dem in Zukunft wohl alle erdenklichen Anwendungen erwachsen werden.

Die persönliche Nachvollziehbarkeit schwieriger Berechnungen und vielfältiger logischer Abhängigkeiten wird die berechtigte Sorge gegenüber dem *schwarzen Kasten* EDV in weiten Bereichen mildern können.

Die zukünftige Hardware-Entwicklung wird wohl auf eine weitere Vergrößerung des verfügbaren Speicherplatzes, einer Steigerung des Programmierkomforts im Hinblick auf **Makrobefehle** mit Anlehnung an *höhere Programmiersprachen* und einer Verbesserung der Ausgabemöglichkeiten im Hinblick auf *alphanumerische Ausdrücke* hinzielen. Das vorhandene Software-Potential wird immer stärker eine *Programmkompatibilität* zu den Folgegenerationen erfordern.

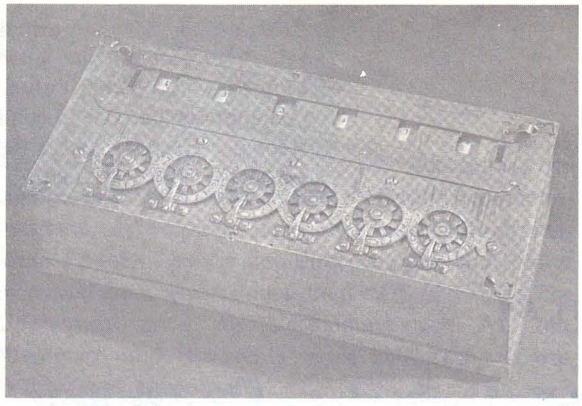


**Bild 1** Rechenmaschine von Blaise Pascal aus dem Jahr 1642

### Literatur


[1] *Ameling, W., Lange, O.:* Stationen auf dem Wege zur Elektronischen Datenverarbeitung, Hochschuljahrbuch Alma Mater Aquensis, Bd. XIV, 1976/77.

[2] *Baxandall, D., Pugh, J.:* Calculating Machines and Instruments Science Museum, London, ISBN 090 1805.14.9.

[3] *Jordan, W.:* Machine arithmetica in qua non additio tantum et subtractio sed et multiplicatio nullo, divisio vero paene nullo animi labore peragantur. (Arithmetische Maschine, mit der man nicht nur Additionen und Subtraktionen, sondern auch die Multiplikation mühelos und die Division fast mühelos vollziehen kann.) Aus Zeitschrift für Vermessungswesen Bd. XXVI, 1877, H. 10, S. 307. Quelle: Niedersächsische Landesbibliothek Hannover (Leibniz-Archiv).


**Vierspezies-Rechner:** Beherrscht nur die vier Grundrechenarten, heute manchmal ergänzt durch Prozent- oder Wurzelautomatik, teilweise noch mit Konstantenautomatik und Speicher versehen.

**Monolithischer Schaltkreis:** Integrierter Schaltkreis (IS bzw. IC: *Integrated Circuit*).

**Nanosekunde:** 0,000 000 001 s bzw. $10^{-9}$ s. Wichtige Bezeichnungen für Bruchteile sind Milli ($10^{-3}$), Mikro ($10^{-6}$), Nano ($10^{-9}$), Piko ($10^{-12}$).

**Externe Speicher:** Speichermedien außerhalb der eigentlichen Rechner zur Aufnahme größerer Mengen von Daten und Programmen (z.B. Magnetstreifen, Magnetkarten, Magnetbandkassetten, Magnetplatten).

**Hardware:** Sämtliche Bauteile, Geräte, Anschaltungen, also alle „greifbaren" Bestandteile eines EDV-Systems.

**Software:** Gesamtheit aller Programme und Programmierungshilfen, durch die eine EDV-Maschine erst zum sinnvoll arbeitenden EDV-System wird.

**Innovation:** lat. Erneuerung, Verjüngung; engl. Neuerung. Gemeint ist die Einführung neuer Bauelemente, Geräte, Programme, Systeme etc. am Markt.

**Makrobefehl:** Wenn eine höhere Programmiersprache „Makro-fähig" ist, ergibt sich folgende Möglichkeit: Häufig wiederkehrende Programmteile werden nur einmal geschrieben und mit einem *Namen* gekennzeichnet. Im Gesamtprogramm braucht dann nur noch dieser Name – der Makrobefehl – zu stehen.

Bei Leistungstests wird der Mehrfach-Wahl-Antwort-Typ bei der Aufgabenstellung häufig benutzt. Bei dieser Antwortform muß unter mehreren vorgegebenen Antwortmöglichkeiten die richtige herausgesucht werden. Die Beliebtheit dieser Multiple-Choice-Tests ist einerseits durch ihre zeitsparende Durchführung und Auswertung begründet, andererseits sind Multiple-Choice-Tests oft von höherer Objektivität als andere Formen der Leistungsmessung und zudem läßt sich die Qualität eines solchen Tests durch gewisse Meßgrößen erfassen, die bei Analyse nach Durchführung des Tests berechnet werden können.

V. Lehmann / H. Ensel

# Korrektur von Multiple-Choice-Tests mit dem TI-59

Das hier vorgestellte Programm führt für einen Multiple-Choice-Test die vollständige Testauswertung durch. Diese umfaßt neben der Testkorrektur und dem Ausdruck der Ergebnisse auch die **Distraktorenanalyse** einschließlich Ausdruck gewisser statistisch relevanter Größen, die zur Beurteilung der Qualität des verwendeten Tests benötigt werden.

Die Kapazität des Programms ist durch die Speicherzahl des TI-59 beschränkt und darf 20 Testfragen mit höchstens fünf Alternativen, wobei genau eine Antwort richtig ist, nicht übersteigen. Die Anzahl der an dem Test teilnehmenden Personen ist auf 40 begrenzt.

Hauptproblem ist, je 20 Antworten von 40 Personen, also 800 Daten gleichzeitig im Rechner zu speichern und dann die Auswertung durchführen zu lassen. Die Antwortalternativen sind mit 1 bis 5 bezeichnet; 0 bedeutet, daß keine Antwort gegeben wurde. Die Speicherung der 800 Antworten wird dadurch möglich, daß die maximal 20 Antworten jeder Testperson zu zwei ‚Antwortzeilen' zusammengefaßt werden. Entsprechend der 10-stelligen Anzeigekapazität des Rechners können so zweimal 10 Antworten in zwei Registern gespeichert werden. Die ‚Lösungszeile' wird in gleicher Weise zusammengefaßt und eingegeben.

Das Programm ist durch einen Trick ganz erheblich beschleunigt worden, was an einem einfachen Zahlenbeispiel erläutert sei. Man bildet zur richtigen Lösungszeile das **Fünferkomplement**; ist beispielsweise $L = 321$ die richtige Lösungszeile, so geht man über zu $\bar{L} = 555 - 321 = 234$. Die Summe aus dem *Komplement* $\bar{L}$ und der gegebenen Antwort $A$ enthält an der Stelle, die einer richtigen Antwort entspricht, stets eine 5. Ist beispielsweise $A = 311$ die gegebene Antwortzeile, so ergibt sich $\bar{L} + A = 234 + 311 = 545$. Nur die zweite Antwort ist falsch, entsprechend ist nur die zweite Ziffer

**Distraktoren:** Die zu jeder Aufgabe vorgegebenen Falsch-Antworten. Distraktion heißt Trennung, Zerstreuung. Hier bedeutet *Distraktorenanalyse* die Interpretation der Testergebnisse durch Ermittlung statistischer Kennwerte, welche z.B. über Schwierigkeit und Trennschärfe der Testaufgaben für eine bestimmte Schülerstichprobe eine Aussage machen.

**Fünferkomplement:** Komplement bedeutet Ergänzung. In der Mathematik ist z.B. das *Einerkomplement* die Ergänzung einer Ziffer auf den Wert 1, d.h. das Einerkomplement von 1 ist 0, das von 0 ist 1. Entsprechend ist das *Fünferkomplement* die Ergänzung auf den Wert 5.

der Summe $\bar{L} + A$ von 5 verschieden. Durch diese Umformung braucht im Korrekturteil immer nur gegen 5 getestet zu werden, und der Inhalt des Testregisters muß nicht ständig geändert werden. Die Folge ist eine erhebliche Zeitersparnis bei der Durchführung des Korrekturprogramms.

Das Gesamtprogramm besteht aus neun Einzelprogrammen, die auf mehrere **Magnetkarten** verteilt sind:

1. Eingabeteil mit Berechnung der Punktesummen (Karten 1 und 2)
2. Häufigkeitsverteilung der Punktesummen (Karte 3a)
3. Durchschnitt, **Standardabweichung** und **Reliabilitätskoeffizient** (Karte 3b)
4. Antworttabelle (Karte 4a)
5. Vorbereitungsprogramm für die Distraktorenanalyse und die **Schwierigkeitsindizes** (Karte 4b)
6. Distraktorenanalyse (Karte 5a)
7. Schwierigkeitsindizes (Karte 5b)
8. Vorbereitungsprogramm für die Trennschärfeindizes (Karte 6a)
9. Trennschärfeindizes (Karte 6b)

Aus Platzgründen wird nur das erste Programm (Eingabeteil und Berechnung der Punktesummen) mit **Programmablaufplan** und Auflistung des Programms vorgestellt. Die anderen Programme werden erläutert aber nicht aufgelistet*).

Das Programm decodiert die Antwortzeilen während der Testauswertung und speichert die den einzelnen Antworten zugeordneten Zwischenergebnisse als Nachkommateil im Schutzstellenbereich des Rechners. Dadurch genügen 90 Speicher, um die Rechenergebnisse für spätere statistische Auswertungen zu sichern, ohne daß die eingegebenen Daten für weitere Rechnungen verloren gehen. Der TI-59 rechnet intern mit 13 Stellen, obwohl nur 10 Stellen in der Anzeige sichtbar sind. Der nicht angezeigte Bereich wird als ‚Schutzstellenbereich' bezeichnet.

Die Anfangsverteilung des Rechners ist 479.59, das bedeutet, daß 480 Programmspeicherschritte und 60 Datenspeicher zu Verfügung stehen. Da vor Beginn der Eingabe der Antwortzeilen nicht alle 90 Datenspeicher benötigt werden, die für das Korrekturprogramm erforderlich sind, kann die zweite Hälfte

*) Interessierte Leser mögen sich mit dem erstgenannten der beiden Autoren in Verbindung setzen:
*Volkhart Lehmann,* Simmerner Str. 51, 5400 Koblenz

```
--------------------
MULTIPLE CHOICE
--------------------
TESTKORREKTUR
--------------------
KLASSE
--------------------
DATUM
--------------------
LEHRER
--------------------
TESTNAME
--------------------

--------------------
```

**Bild 1** Ausgedruckte Überschriften

des Programmspeichers für das Ausdrucken der Überschriften (**Bild 1**) verwendet werden. Nach Abarbeiten des Programms springt der Rechner durch einen **Reset-Befehl** an den Anfang des Programms und schaltet automatisch auf die Verteilung 239.89 um. Die für das Ausdrucken der Überschriften erforderlichen Codezahlen (**Bild 2**) werden zu Beginn über die Magnetkarten eingelesen und sind in den Speichern R10 bis R37 enthalten.

**Magnetkarte:** Plastikstreifen mit magnetischer Beschichtung (wie beim „Tonband") zur Speicherung von Programmen und Daten (zweimal 240 Programmschritte beim TI-59).

**Standardabweichung:** Bei Meßreihen von wiederholten Beobachtungen kann man in der Regel die „Streuung" um einen *Mittelwert* angeben. Die Quadratwurzel aus dieser Streuung wird Standardabweichung genannt.

**Reliabilität:** (engl.: *Reliability*). Zuverlässigkeit bei Tests oder Meßreihen.

**Koeffizient:** Vorzahl oder Beizahl; eine Zahl, mit der eine unbekannte oder veränderliche Größe multipliziert wird.

**Indizes:** Mehrzahl von Index (Anzeiger).

**Programmablaufplan:** Zur grafischen Darstellung von Programmen und Datenbewegungen werden *Flußdiagramme* verwendet (s. dazu Beitrag *Schneider:* Grafische Darstellung von Programmabläufen). Man unterscheidet dabei *Programmablaufplan* und *Datenflußplan.*

**Reset:** Rücksetzen. Kommando zur Erzeugung eines definierten Anfangszustands in einem Datenverarbeitungssystem.

```
        0.   00
        0.   01
       11.   02
        8.   03
        0.   04
5555555555.  05
5555555555.  06
3647303017.  07
        0.   08
        0.   09
2020202020.  10   - - - - -
3041273724.  11   M U L T I
3327170015.  12   P L E   C
2332241517.  13   H O I C E
3717363726.  14   T E S T K
3235351726.  15   O R R E K
3741350000.  16   T U R
2627133636.  17   K L A S S
1700000000.  18   E
        0.   19
1613374130.  20   D A T U M
        0.   21
        0.   22
2717233517.  23   L E H R E
3500000000.  24   R
        0.   25
3717363731.  26   T E S T N
1330170000.  27   A M E
        0.   28
        0.   29
        0.   30
        0.   31
2732173641.  32   L O E S U
3122364617.  33   N G S Z E
2427173100.  34   I L E N
1331374332.  35   A N T W O
3537461724.  36   R T Z E I
2717310000.  37   L E N
        0.   38
        0.   39
        0.   40
        0.   41
```

**Bild 2** Codeliste für das Ausdrucken der Überschriften

## Vorstellung der einzelnen Programme

Das erste Programm *‚Eingabeteil mit Berechnung der Punktesummen'* umfaßt Eingabe der Lösungs- und Antwortzeilen, sowie die Berechnung der absoluten und relativen Punktesumme für jeden Schüler, d.h. es werden nebeneinander die absolute Anzahl der richtigen Antworten und der prozentuale Anteil mit der Schülernummer am Rand ausgedruckt (**Bild 3**).

Zur Übersicht des Programms soll ein *Programmablaufplan* (**Bild 4**) gegeben werden. Für dieses Programm benötigt man eine Magnetkarte, die auf beiden Seiten beschriftet ist, sowie eine weitere Magnetkarte auf der Codezahlen (**Bild 2**) für Überschriften gespeichert sind.

Nach Eingabe der Aufgabenzahl *p* erfolgt der Ausdruck (**Bild 1**). Bei dem Ausdruck der zweiten Antwortzeile, sowie bei der Punktesumme, wird die jeweilige Schülernummer *i* am Rand mit ausgedruckt. (**Bild 3**) Dies wird durch das Unterprogramm SBR 191 ermöglicht, in dem die Druckercodezahlen für die Schülernummer *i* ermittelt und mit der Steueroperation Op 04 im Druckregister gespeichert werden. Wird nun die Antwortzeile oder die Punktesumme über Op 06 ausgedruckt, so wird die Schülernummer am Rand mit ausgedruckt.

Im weiteren ist auf der zweiten Magnetkarte in R05 und R06 jeweils die Zahl 5555555555 gespeichert (**Bild 2**). Diese Zahlen sind für den bereits erläuterten Trick zur Beschleunigung des Programms erforderlich. Das Programm ist unter **Bild 5** aufgelistet.

Der zweite Programmteil berechnet die *‚Häufigkeitsverteilung'*. Hier wird zu jeder möglichen Punktzahl die Anzahl der Schüler berechnet, die diese Punktzahl erreicht haben. Die jeweilige Punktzahl wird als Nachkommateil der Anzahl der Punkte mit ausgedruckt (**Bild 6**). Zu Anfang des Programms wird eine Sortierung der Register, in denen sich die Antwortzeilen befinden (R10, R11, R12 ...) im Unter-

```
LOESUNGSZEILEN
 2233114131.
 2124413224.
--------------------
ANTWORTZEILEN
 2213114224.
 3121313224.      01

 2213112134.
 3121413222.      02

 2213114234.
 3312313224.      03

                        PUNKTESUMMEN
 2213114231.                  13.065   01
 3312313221.      04          14.070   02
                              12.060   03
 2234114131.                  12.060   04
 2134114122.      05          14.070   05
                   ↑                    ↑
                   └──── Schülernummer ────┘
```

**Bild 3** Beispiel für das Ermitteln der Punktesumme für die einzelnen Schüler

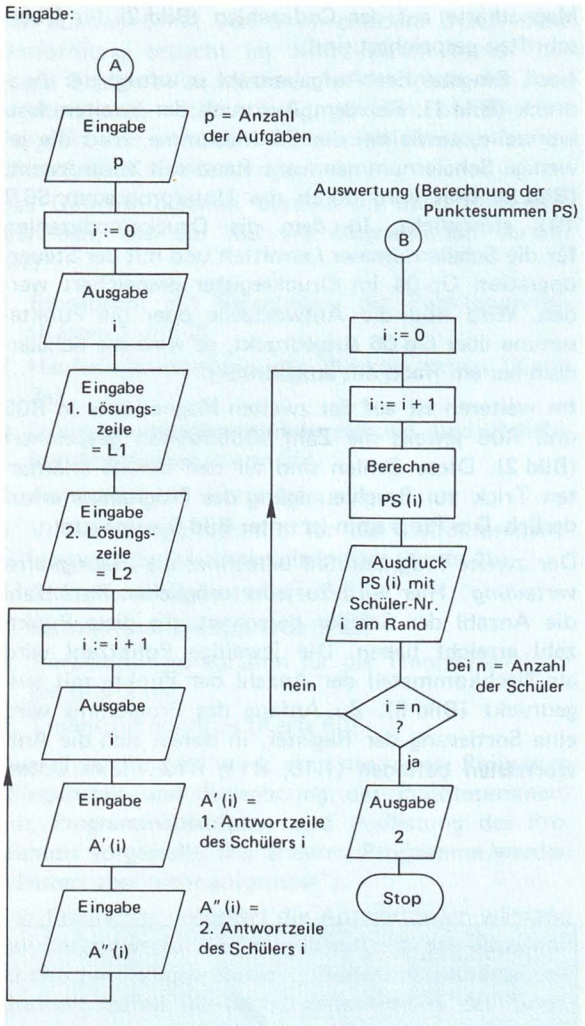



**Bild 4** Programmablaufplan für die Berechnung der Punktesummen

programm SBR 000 vorgenommen. Die Sortierung erfolgt nach der Anzahl der richtigen Antworten, die als Nachkommateil in R11, R13 ... gespeichert sind. Das Programm sortiert also die Speicherinhalte der erwähnten ungeraden Speicher entsprechend dem Nachkommateil und beim Austauschen tauscht es die jeweiligen vorherigen geraden Speicher mit aus. So befinden sich schließlich die Antwortzeilen des besten Schülers in R10 und R11 usw. ...

Dieser Teil des Programms ist auch notwendig für die Berechnung der Trennschärfeindizes. Wenn man die Häufigkeitsverteilung nicht durchführen möchte, jedoch die Berechnung der Trennschärfeindizes, so ist es möglich, nur den Programmteil des Sortierens ohne Berechnung der Häufigkeitsverteilung durchzuführen.

Im dritten Teil des Programms lassen sich der *Durchschnitt* und die *Standardabweichung* der Häufigkeitsverteilung berechnen (**Bild 7**). Dies läßt sich mit dem in dem Rechner eingebautem Modul 1 sehr leicht realisieren, das ein Programm zur Berechnung solcher statistischen Werte hat. Damit läßt sich auch der sogenannte *Reliabilitätskoeffizient* berechnen. Er stellt ein Maß für die Zuverlässigkeit des Tests dar, d.h. er macht eine Aussage, inwieweit der Test reproduzierbar ist.

*Lienert**) schlägt folgende Formel zur Berechnung des Reliabilitätskoeffizienten vor:

$$r = 1 - \frac{n - M}{m - s^2}$$

mit $n$ = Anzahl der Aufgaben, $M$ = Arithmetisches Mittel, $m$ = Anzahl der Wahlmöglichkeiten pro Aufgabe, $s^2$ = Varianz.

*Lienert* empfiehlt diese Formel nur unter folgenden Voraussetzungen, die ein Test enthalten muß:

1. Die Aufgaben sind einander gleichgestellt,
2. Alle Schüler konnten prinzipiell alle Aufgaben in Angriff nehmen,
3. Die Ratewahrscheinlichkeit für alle Aufgaben ist gleich groß.

Problem 1 wird in unserem Test als Voraussetzung akzeptiert. Ob der zweite Punkt seine Gültigkeit hat, muß bei den jeweiligen Tests einzeln untersucht werden. Der dritte Punkt läßt sich mit Hilfe der Distraktorenanalyse untersuchen.

Der vierte Teil des Programms ermittelt eine *‚Antworttabelle'* (**Bild 8**). Es wird eine Tabelle aufgestellt, aus der man ablesen kann, wieviele Schüler die Alternative A, B, C, D oder E angekreuzt haben. Die jeweilige Aufgabennummer wird am Rand mit ausgedruckt. Für jeden Schüler wird bei der jeweiligen Aufgabe die Lösungsziffer aus der 10-stelligen Antwortzeile isoliert, und durch INV log in eine Platzziffer verwandelt, die in die jeweilige Tabellenzeile dazuaddiert wird.

*) *Lienert:* Testaufbau und Testanalyse (S. 228).

| | | |
|---|---|---|
| 000 | 69 | OP |
| 001 | 17 | 17 |
| 002 | 01 | 1 |
| 003 | 76 | LBL |
| 004 | 13 | C |
| 005 | 42 | STO |
| 006 | 00 | 00 |
| 007 | 65 | × |
| 008 | 02 | 2 |
| 009 | 85 | + |
| 010 | 08 | 8 |
| 011 | 95 | = |
| 012 | 42 | STO |
| 013 | 01 | 01 |
| 014 | 43 | RCL |
| 015 | 00 | 00 |
| 016 | 91 | R/S |
| 017 | 72 | ST* |
| 018 | 01 | 01 |
| 019 | 99 | PRT |
| 020 | 91 | R/S |
| 021 | 69 | OP |
| 022 | 21 | 21 |
| 023 | 72 | ST* |
| 024 | 01 | 01 |
| 025 | 71 | SBR |
| 026 | 01 | 01 |
| 027 | 91 | 91 |
| 028 | 73 | RC* |
| 029 | 01 | 01 |
| 030 | 69 | OP |
| 031 | 06 | 06 |
| 032 | 69 | OP |
| 033 | 20 | 20 |
| 034 | 69 | OP |
| 035 | 21 | 21 |
| 036 | 98 | ADV |
| 037 | 61 | GTO |
| 038 | 00 | 00 |
| 039 | 14 | 14 |
| 040 | 76 | LBL |
| 041 | 12 | B |
| 042 | 69 | OP |
| 043 | 30 | 30 |
| 044 | 43 | RCL |
| 045 | 00 | 00 |
| 046 | 42 | STO |
| 047 | 04 | 04 |
| 048 | 71 | SBR |
| 049 | 00 | 00 |
| 050 | 80 | 80 |
| 051 | 55 | ÷ |
| 052 | 01 | 1 |
| 053 | 00 | 0 |
| 054 | 95 | = |
| 055 | 42 | STO |
| 056 | 03 | 03 |
| 057 | 22 | INV |
| 058 | 59 | INT |
| 059 | 65 | × |
| 060 | 01 | 1 |
| 061 | 00 | 0 |
| 062 | 95 | = |
| 063 | 32 | X:T |
| 064 | 05 | 5 |
| 065 | 22 | INV |
| 066 | 67 | EQ |
| 067 | 00 | 00 |
| 068 | 71 | 71 |
| 069 | 69 | OP |
| 070 | 22 | 22 |
| 071 | 29 | CP |
| 072 | 43 | RCL |
| 073 | 03 | 03 |
| 074 | 59 | INT |
| 075 | 22 | INV |
| 076 | 67 | EQ |
| 077 | 00 | 00 |
| 078 | 51 | 51 |
| 079 | 92 | RTN |
| 080 | 42 | STO |
| 081 | 07 | 07 |
| 082 | 00 | 0 |
| 083 | 69 | OP |
| 084 | 04 | 04 |
| 085 | 69 | OP |
| 086 | 05 | 05 |
| 087 | 43 | RCL |
| 088 | 08 | 08 |
| 089 | 59 | INT |
| 090 | 22 | INV |
| 091 | 44 | SUM |
| 092 | 05 | 05 |
| 093 | 43 | RCL |
| 094 | 09 | 09 |
| 095 | 22 | INV |
| 096 | 44 | SUM |
| 097 | 06 | 06 |
| 098 | 01 | 1 |
| 099 | 00 | 0 |
| 100 | 42 | STO |
| 101 | 01 | 01 |
| 102 | 43 | RCL |
| 103 | 05 | 05 |
| 104 | 85 | + |
| 105 | 73 | RC* |
| 106 | 01 | 01 |
| 107 | 95 | = |
| 108 | 71 | SBR |
| 109 | 00 | 00 |
| 110 | 51 | 51 |
| 111 | 69 | OP |
| 112 | 21 | 21 |
| 113 | 43 | RCL |
| 114 | 06 | 06 |
| 115 | 85 | + |
| 116 | 73 | RC* |
| 117 | 01 | 01 |
| 118 | 95 | = |
| 119 | 71 | SBR |
| 120 | 00 | 00 |
| 121 | 51 | 51 |
| 122 | 43 | RCL |
| 123 | 02 | 02 |
| 124 | 55 | ÷ |
| 125 | 01 | 1 |
| 126 | 00 | 0 |
| 127 | 00 | 0 |
| 128 | 95 | = |
| 129 | 74 | SM* |
| 130 | 01 | 01 |
| 131 | 00 | 0 |
| 132 | 42 | STO |
| 133 | 02 | 02 |
| 134 | 69 | OP |
| 135 | 21 | 21 |
| 136 | 97 | DSZ |
| 137 | 04 | 04 |
| 138 | 01 | 01 |
| 139 | 02 | 02 |
| 140 | 01 | 1 |
| 141 | 01 | 1 |
| 142 | 42 | STO |
| 143 | 02 | 02 |
| 144 | 43 | RCL |
| 145 | 02 | 02 |
| 146 | 75 | - |
| 147 | 09 | 9 |
| 148 | 95 | = |
| 149 | 55 | ÷ |
| 150 | 02 | 2 |
| 151 | 95 | = |
| 152 | 71 | SBR |
| 153 | 01 | 01 |
| 154 | 93 | 93 |
| 155 | 73 | RC* |
| 156 | 02 | 02 |
| 157 | 22 | INV |
| 158 | 59 | INT |
| 159 | 55 | ÷ |
| 160 | 01 | 1 |
| 161 | 00 | 0 |
| 162 | 55 | ÷ |
| 163 | 43 | RCL |
| 164 | 08 | 08 |
| 165 | 22 | INV |
| 166 | 59 | INT |
| 167 | 85 | + |
| 168 | 73 | RC* |
| 169 | 02 | 02 |
| 170 | 22 | INV |
| 171 | 59 | INT |
| 172 | 65 | × |
| 173 | 01 | 1 |
| 174 | 00 | 0 |
| 175 | 00 | 0 |
| 176 | 95 | = |
| 177 | 58 | FIX |
| 178 | 03 | 03 |
| 179 | 69 | OP |
| 180 | 06 | 06 |
| 181 | 22 | INV |
| 182 | 58 | FIX |
| 183 | 02 | 2 |
| 184 | 44 | SUM |
| 185 | 02 | 02 |
| 186 | 97 | DSZ |
| 187 | 00 | 00 |
| 188 | 01 | 01 |
| 189 | 44 | 44 |
| 190 | 91 | R/S |
| 191 | 43 | RCL |
| 192 | 00 | 00 |
| 193 | 55 | ÷ |
| 194 | 01 | 1 |
| 195 | 00 | 0 |
| 196 | 75 | - |
| 197 | 59 | INT |
| 198 | 42 | STO |
| 199 | 04 | 04 |
| 200 | 95 | = |
| 201 | 65 | × |
| 202 | 01 | 1 |
| 203 | 00 | 0 |
| 204 | 85 | + |
| 205 | 01 | 1 |
| 206 | 95 | = |
| 207 | 32 | X:T |
| 208 | 08 | 8 |
| 209 | 32 | X:T |
| 210 | 22 | INV |
| 211 | 77 | GE |
| 212 | 02 | 02 |
| 213 | 17 | 17 |
| 214 | 85 | + |
| 215 | 02 | 2 |
| 216 | 95 | = |
| 217 | 42 | STO |
| 218 | 03 | 03 |
| 219 | 69 | OP |
| 220 | 24 | 24 |
| 221 | 01 | 1 |
| 222 | 00 | 0 |
| 223 | 00 | 0 |
| 224 | 49 | PRD |
| 225 | 04 | 04 |
| 226 | 43 | RCL |
| 227 | 04 | 04 |
| 228 | 44 | SUM |
| 229 | 03 | 03 |
| 230 | 04 | 4 |
| 231 | 22 | INV |
| 232 | 28 | LOG |
| 233 | 49 | PRD |
| 234 | 03 | 03 |
| 235 | 43 | RCL |
| 236 | 03 | 03 |
| 237 | 69 | OP |
| 238 | 04 | 04 |
| 239 | 92 | RTN |
| 240 | 43 | RCL |
| 241 | 10 | 10 |
| 242 | 69 | OP |
| 243 | 01 | 01 |
| 244 | 69 | OP |
| 245 | 02 | 02 |
| 246 | 69 | OP |
| 247 | 03 | 03 |
| 248 | 69 | OP |
| 249 | 04 | 04 |
| 250 | 69 | OP |
| 251 | 05 | 05 |
| 252 | 69 | OP |
| 253 | 00 | 00 |
| 254 | 92 | RTN |
| 255 | 01 | 1 |
| 256 | 42 | STO |
| 257 | 03 | 03 |
| 258 | 71 | SBR |
| 259 | 02 | 02 |
| 260 | 40 | 40 |
| 261 | 73 | RC* |
| 262 | 02 | 02 |
| 263 | 69 | OP |
| 264 | 01 | 01 |
| 265 | 69 | OP |
| 266 | 22 | 22 |
| 267 | 73 | RC* |
| 268 | 02 | 02 |
| 269 | 69 | OP |
| 270 | 02 | 02 |
| 271 | 69 | OP |
| 272 | 22 | 22 |
| 273 | 73 | RC* |
| 274 | 02 | 02 |
| 275 | 69 | OP |
| 276 | 03 | 03 |
| 277 | 69 | OP |
| 278 | 22 | 22 |
| 279 | 69 | OP |
| 280 | 05 | 05 |
| 281 | 87 | IFF |
| 282 | 00 | 00 |
| 283 | 03 | 03 |
| 284 | 03 | 03 |
| 285 | 97 | DSZ |
| 286 | 03 | 03 |
| 287 | 02 | 02 |
| 288 | 58 | 58 |
| 289 | 25 | CLR |
| 290 | 91 | R/S |
| 291 | 44 | SUM |
| 292 | 08 | 08 |
| 293 | 99 | PRT |
| 294 | 91 | R/S |
| 295 | 44 | SUM |
| 296 | 09 | 09 |
| 297 | 99 | PRT |
| 298 | 86 | STF |
| 299 | 00 | 00 |
| 300 | 61 | GTO |
| 301 | 02 | 02 |
| 302 | 55 | 55 |
| 303 | 00 | 0 |
| 304 | 42 | STO |
| 305 | 02 | 02 |
| 306 | 03 | 3 |
| 307 | 03 | 3 |
| 308 | 04 | 4 |
| 309 | 01 | 1 |
| 310 | 03 | 3 |
| 311 | 01 | 1 |
| 312 | 02 | 2 |
| 313 | 06 | 6 |
| 314 | 03 | 3 |
| 315 | 07 | 7 |
| 316 | 69 | OP |
| 317 | 01 | 01 |
| 318 | 01 | 1 |
| 319 | 07 | 7 |
| 320 | 03 | 3 |
| 321 | 06 | 6 |
| 322 | 04 | 4 |
| 323 | 01 | 1 |
| 324 | 03 | 3 |
| 325 | 00 | 0 |
| 326 | 03 | 3 |
| 327 | 00 | 0 |
| 328 | 69 | OP |
| 329 | 02 | 02 |
| 330 | 01 | 1 |
| 331 | 07 | 7 |
| 332 | 03 | 3 |
| 333 | 01 | 1 |
| 334 | 00 | 0 |
| 335 | 00 | 0 |
| 336 | 00 | 0 |
| 337 | 00 | 0 |
| 338 | 00 | 0 |
| 339 | 00 | 0 |
| 340 | 69 | OP |
| 341 | 03 | 03 |
| 342 | 09 | 9 |
| 343 | 81 | RST |
| 344 | 76 | LBL |
| 345 | 11 | A |
| 346 | 29 | CP |
| 347 | 42 | STO |
| 348 | 04 | 04 |
| 349 | 75 | - |
| 350 | 01 | 1 |
| 351 | 00 | 0 |
| 352 | 95 | = |
| 353 | 77 | GE |
| 354 | 03 | 03 |
| 355 | 73 | 73 |
| 356 | 50 | I×I |
| 357 | 22 | INV |
| 358 | 28 | LOG |
| 359 | 35 | 1/X |
| 360 | 65 | × |
| 361 | 43 | RCL |
| 362 | 05 | 05 |
| 363 | 95 | = |
| 364 | 59 | INT |
| 365 | 42 | STO |
| 366 | 05 | 05 |
| 367 | 25 | CLR |
| 368 | 42 | STO |
| 369 | 06 | 06 |
| 370 | 61 | GTO |
| 371 | 03 | 03 |
| 372 | 90 | 90 |
| 373 | 94 | +/- |
| 374 | 85 | + |
| 375 | 01 | 1 |
| 376 | 00 | 0 |
| 377 | 95 | = |
| 378 | 22 | INV |
| 379 | 28 | LOG |
| 380 | 35 | 1/X |
| 381 | 65 | × |
| 382 | 43 | RCL |
| 383 | 06 | 06 |
| 384 | 95 | = |
| 385 | 59 | INT |
| 386 | 86 | STF |
| 387 | 02 | 02 |
| 388 | 42 | STO |
| 389 | 06 | 06 |
| 390 | 43 | RCL |
| 391 | 04 | 04 |
| 392 | 55 | ÷ |
| 393 | 01 | 1 |
| 394 | 00 | 0 |
| 395 | 00 | 0 |
| 396 | 95 | = |
| 397 | 42 | STO |
| 398 | 08 | 08 |
| 399 | 22 | INV |
| 400 | 87 | IFF |
| 401 | 02 | 02 |
| 402 | 02 | 02 |
| 403 | 58 | 58 |
| 404 | 75 | - |
| 405 | 93 | . |
| 406 | 01 | 1 |
| 407 | 95 | = |
| 408 | 42 | STO |
| 409 | 09 | 09 |
| 410 | 44 | SUM |
| 411 | 06 | 06 |
| 412 | 61 | GTO |
| 413 | 02 | 02 |
| 414 | 58 | 58 |

**Bild 5** Programmliste „Eingabeteil mit Berechnung der Punktesummen"

```
HAEUFIGKEITSV.
        0.20
        1.19
        2.18
        7.17
        0.16
        6.15
        5.14
        5.13
        4.12
        0.11
        0.10
        1.09
        0.08
        0.07
        0.06
        0.05
        0.04
        0.03
        0.02
        0.01
        0.00
        ↑ ↑
   Punktzahl
Anzahl der Schüler
```

**Bild 6** Zuordnung der Ergebnisse von 31 Schülern zu den erreichbaren Punktzahlen

```
A.-TABELLE
 A B C D E
   29020000.   01
   31000000.   02
  900200200.   03
  101260300.   04
 3000000100.   05
 2803000000.   06
  201012700.   07
 1614000100.   08
  204250000.   09
 2003050300.   10
  615070300.   11
 2301020500.   12
  224040100.   13
  507041500.   14
  201111700.   15
 2802010000.   16
  100280200.   17
  518080000.   18
  722020000.   19
  808011400.   20
              ↑
   Aufgabennummer
```

**Bild 8** Antworttabelle

```
DISTRAKTORENANALYSE
        0.01    01
          0.    02
        1.06    03
        1.03    04
        0.01    05
        0.02    06
        0.02    07
        2.08    08
        1.03    09
        1.06    10
        2.08    11
        1.04    12
        1.04    13
        2.08    14
        1.07    15
        0.02    16
        0.02    17
        1.07    18
        1.05    19
        2.09    20
                ↑
   Aufgabennummer
```

**Bild 9** Distraktorenanalyse (vgl. Text)

```
DURCHSCHNITT
      14.71

STANDARDABWEICHUNG
       2.28

RELIABIL.KOEFF.
       0.74
```

**Bild 7** Auswertungen des mit Bild 6 abgedruckten Ergebnisses

Der fünfte Teil des Programms druckt keinen Wert aus. Er dient lediglich zur *Vorbereitung der Distraktorenanalyse und der Berechnung der Schwierigkeitsindizes.* Hier wird für jede Aufgabe ermittelt, wieviele Schüler die jeweilige Aufgabe richtig beantwortet haben. Die einzelnen Werte werden als Nachkommateil im Schutzstellenbereich des Rechners in R10, R12, R14 usw. eingespeichert.

Der sechste Teil dient der Ausführung der *‚Distraktorenanalyse'* (**Bild 9**). Dieser Teil ermittelt, ob die Falschantworten (*Distraktoren*) eine genügende Attraktivität aufweisen. Dies ist notwendig, um eine Verschiebung der Ratewahrscheinlichkeit zugunsten der richtigen Lösung auszuschließen. Über die Frage, nach welchen Kriterien brauchbare von unbrauchbaren Distraktoren zu trennen sind, schlagen *Seelig* und *Kuschmann/Zwirtz* folgendes Verfahren vor:

Entfallen bei drei Distraktoren pro *Item* (Aufgabe) mehr als die Hälfte aller falschen Antworten auf einen Distraktor, so ist dieser Distraktor zu attraktiv. Entfallen weniger als 10% aller falschen Lösungen auf einen Distraktor, so ist der Distraktor zu unattraktiv.*)

Das Programm gibt für jede Aufgabe nebeneinander zwei Zahlen an, wobei im Intervall dieser beiden Zahlen ein Distraktor der jeweiligen Aufgabe den gestellten Ansprüchen genügt, d.h. der erste Wert gibt 10% der falschen Lösungen dieser Aufgabe, der zweite Wert 50% der falschen Lösungen an. Bei jedem Ausdruck wird hier die jeweilige Aufgabennummer am Rand mit ausgedruckt. Mit diesen Werten kann man mit Hilfe der Antworttabelle die Analyse durchführen. Wir haben es bei der Auflistung dieser Werte gelassen, da eine endgültige Analyse durch den Rechner zu zeitraubend wäre.

*) *Kuschmann, W./Zwirtz, H.:* Objektive Leistungsmessung in der Schulpraxis. Neue Unterrichtspraxis 1975. Heft 3 S. 164.

Im siebten Programm wird für jede Aufgabe ein *‚Schwierigkeitsindex'* berechnet, mit dessen Hilfe man nicht nur eine Aussage über das betreffende Item erhält, als vielmehr auch ein Maß bezüglich des Lernziels in dem betreffenden Lernstoff. *Lienert* schlägt hierzu folgende Formel vor:

$$P = 100 \frac{N_r - \frac{N_f}{m-1}}{N}$$

mit $N_r$ = Zahl der Schüler, die die Aufgabe richtig beantworteten, $N_f$ = Zahl der Schüler, die die Aufgabe falsch beantworteten, $m$ = Anzahl der Wahlmöglichkeiten pro Item, $N$ = Anzahl der Schüler.

Daraus ergibt sich, daß hohe Schwierigkeitsindizes für leichte und niedrigere Schwierigkeitsindizes für schwerere Aufgaben stehen. Neben dem jeweiligen Index wird die Aufgabennummer als Nachkommateil mit ausgedruckt (**Bild 10**).

Das achte und neunte Programm befassen sich mit der Ermittlung der *‚Trennschärfe'*. Dieser Index gibt für jede Aufgabe an, inwieweit diese die leistungsfähigen von den weniger leistungsfähigen Schülern trennt. So gibt ein hoher Trennschärfenindex an, daß die Aufgabe von den guten Schülern besser gelöst wurde als von den schlechteren Schülern, wobei ein negativer Index darauf hinweisen würde, daß die schlechteren Schüler die Aufgabe erfolgreicher angingen als die guten Schüler. *Gaude/Teschner**) schlagen folgende Formel vor:

$$T = \frac{N_{ro} - N_{ru}}{\frac{N}{4}} \times 100$$

mit $N_{ro}$ = Richtige Lösungen für das betreffende Item in der oberen Extremgruppe (25 % des am Test teilnehmenden Schüler), $N_{ru}$ = Richtige Lösungen für das betreffende Item in der unteren Extremgruppe, $N$ = Gesamtzahl der teilnehmenden Schüler.

Der Trennschärfeindex setzt die Sortierung der Antwortzeilen (vgl. Häufigkeitsverteilung) voraus. Sie wird in gleicher Form wie der Schwierigkeitsindex ausgedruckt (**Bild 11**).

*) *Gaude, P./Teschner, W.-P.:* Objektivierte Leistungsmessung in der Schule, Frankfurt a.M. 1973. 3. Auflage Seite 112.

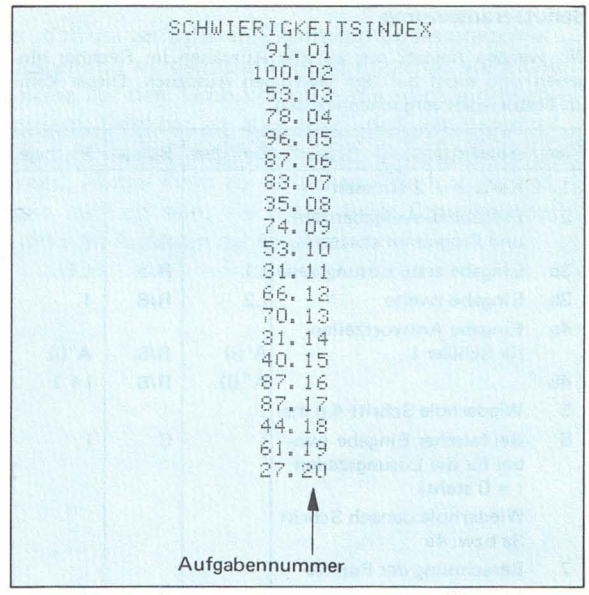



**Bild 10** Schwierigkeitsindex (vgl. Text)

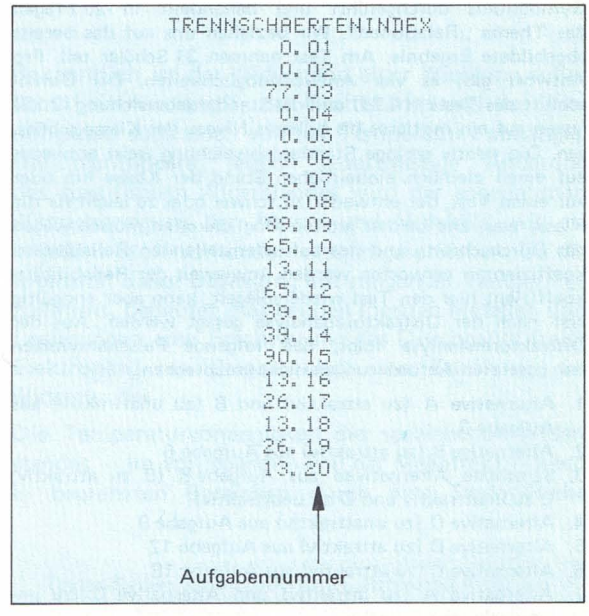



**Bild 11** Trennschärfenindex (vgl. Text)

## Benutzeranweisung

Wir werden hierbei nur auf die Ausgaben im Rechner eingehen und nicht auf den jeweiligen Ausdruck. Dieser kann im Testbeispiel verglichen werden.

| Nr. | Anweisung | Eingabe | Befehl | Anzeige |
|---|---|---|---|---|
| 1 | Karte 1 u. 2 einlesen | | | |
| 2 | Eingabe d. Aufgabenzahl und Programm starten | p | A | 0 |
| 3a | Eingabe erste Lösungszeile | L1 | R/S | L1 |
| 3b | Eingabe zweite " | L2 | R/S | 1 |
| 4a | Eingabe Antwortzeilen für Schüler i | A'(i) | R/S | A'(i) |
| 4b | | A"(i) | R/S | i + 1 |
| 5 | Wiederhole Schritt 4 n mal | | | |
| 6 | Bei falscher Eingabe (wobei für die Lösungszeilen i = 0 steht) | i | C | i |
| | Wiederhole danach Schritt 3a bzw. 4a | | | |
| 7 | Berechnung der Punktesummen | | B | 2 |

Für die anderen Programme wird die jeweilige Karte eingelesen und über A das Programm gestartet. Bei der Häufigkeitsverteilung ist es möglich, nur einen Teil des Programmes durchlaufen zu lassen (vgl. Häufigkeitsverteilung). Außer dem ersten Programm können alle anderen Programme beliebig übersprungen werden.

## Testbeispiel

Der hier vorgestellte Test wurde in einer neunten Klasse eines Gymnasiums durchgeführt und behandelte in 20 Fragen das Thema „Relationen". Wir beziehen uns auf das bereits abgebildete Ergebnis. Am Test nahmen 31 Schüler teil. Pro Antwort gibt es vier Antwortmöglichkeiten. Der Durchschnitt des Tests (14.71) und die Standardabweichung (2.28) lassen auf ein mittleres bis höheres Niveau der Klasse schliessen. Die relativ geringe Standardabweichung weist entweder auf einen ziemlich einheitlichen Stand der Klasse hin oder auf einen Test, der entweder zu schwer oder zu leicht für die Klasse war. Die beiden letzten Möglichkeiten müssen wegen des Durchschnitts und des zufriedenstellenden Reliabilitätskoeffizienten verworfen werden. Inwieweit der Reliabilitätskoeffizient hier den Test wiederspiegelt, kann aber endgültig erst nach der Distraktorenanalyse gesagt werden. Aus der Distraktorenanalyse folgt, daß folgende Falschantworten den gesetzten Anforderungen nicht entsprechen:

1. Alternative A (zu attraktiv) und B (zu unattraktiv) aus Aufgabe 3
2. Alternative B (zu attraktiv) aus Aufgabe 6
3. Sämtliche Alternativen aus Aufgabe 8 (B zu attraktiv; C zu unattraktiv und D zu unattraktiv)
4. Alternative D (zu unattraktiv) aus Aufgabe 9
5. Alternative D (zu attraktiv) aus Aufgabe 12
6. Alternative C (zu attraktiv) aus Aufgabe 18
7. Alternative A (zu attraktiv) und Alternative D (zu unattraktiv) aus Aufgabe 19
8. Alternative C (zu unattraktiv) aus Aufgabe 20

Hier zeigt sich, daß relativ viele Distraktoren den gestellten Anforderungen nicht genügen. Besonders auffallend sind die Aufgaben 3, 8 u. 19, die bei erneuter Verwendung des Testes überarbeitet werden sollten, da auch, besonders bei der Aufgabe 8, sich ein überdurchschnittlicher Schwierigkeitsindex zeigt.

Die Schwierigkeitsindizes weisen auf sehr unterschiedliche Aufgabentypen hin; dies zeigen auch die Trennschärfenindizes. Dabei fällt besonders Aufgabe 2 auf, die mit einem Schwierigkeitsindex von 100 zweifelslos zu leicht ist.

In Aufgabe 20 finden wir den geringsten Schwierigkeitsindex (27), der aber ohne weiteres noch vertretbar ist. Insgesamt spiegelt sich in den Schwierigkeitsindizes der respektable Durchschnitt wieder. Man kann hier also von einem durchschnittlich schwierigen Test sprechen.

Aus der Bearbeitung bisher geht hervor, daß man dem Reliabilitätskoeffizienten nur beschränkt seine Gültigkeit hier zusprechen kann, da aus der Distraktorenanalyse hervorging, daß die Ratewahrscheinlichkeit zugunsten der richtigen Alternative nicht bei jeder Aufgabe gleich hoch ist. Dennoch kann man sich ohne weiteres dazu entschließen, den Test nach Überarbeitung wieder zu benutzen, zumal die Analyse keine gravierenden Mängel aufweist.

Der Trennschärfeindex zeigt, daß vier von den 20 gestellten Aufgaben keine genügende Trennschärfe aufweisen. Dies unterstützt die Vermutung, daß das Niveau der Klasse relativ einheitlich ist (vgl. Standardabweichung). Aufgabe 15 zeigt mit dem Index 90 die absolut größte Trennschärfe.

Erstaunlich zeigt sich der relativ geringe Wert von Aufgabe 20, zumal diese den höchsten Schwierigkeitsindex aufweist. Man muß hier auf eine ziemlich hohe Ratewahrscheinlichkeit schließen, obwohl dies aus der Distraktorenanalyse nicht so deutlich hervorgeht.

## Literatur

*Hans Jürgen Becker/Günter Jüngel:* Erarbeitung und Erprobung eines Chemietests (Sekundarstufe 1).

*Gaude, P./Teschner, W.-P.:* Objektivierte Leistungsmessung in der Schule, Frankfurt a.M. 1973 3. Auflage.

*Lienert, G.:* Testaufbau und Testanalyse, Weinheim 1969.

*Kuschmann, W./Zwirtz, H.:* Objektive Leistungsmessung in der Schulpraxis, Neue Unterrichtspraxis, 1975, Heft 3 (Teil 1).

---

Das Programm zur Sortierung der Datenspeicher (vgl. Häufigkeitsverteilung) verdanken wir *Chahriar Bidardel,* Koblenz.

Die Erfahrung lehrt immer wieder, daß die verfügbaren Eigenschaften programmierbarer Taschenrechner nur selten ausgenutzt werden. Das liegt im Einzelfall vielleicht daran, daß einem nie solche Probleme auf den Tisch kommen, die solch ein hochentwickeltes Gerät vollständig fordern. Denkbar ist aber auch, daß der Benutzer sich oder seinem Rechner nicht genug zutraut. Der folgende Beitrag soll unter anderem auch als Anregung dafür dienen, einmal mehr zu versuchen. Wie hier gezeigt, kann dabei ein Drucker besonders nützlich sein, als Helfer beim Schreiben des Programms, zur Dokumentation und zum Auflisten der Ergebnisse.

W.-R. Haberditz

# Anwendung des programmierbaren Taschenrechners TI-59 bei der Temperaturbestimmung an elektrischen Wicklungen

## 1 Vorbetrachtungen

In der Elektrotechnik, der Elektronik und auch auf anderen Gebieten ist es häufig erforderlich, die mittlere Temperatur stromdurchflossener Wicklungen zu bestimmen. Von besonderer Bedeutung ist dies z.B. für die Beurteilung und Festlegung der Grenzdaten bei Netztransformatoren, Maschinen, Motoren, Feld- und Drosselspulen, Elektromagneten, Relais, Glühlampen etc. Aus verschiedenen Gründen – u.a. Kosten, Platzbedarf, Wärmeableitung und Leitungszuführung – kommen festeingebaute Temperaturfühler in der Regel nicht in Frage.

Da das Innere einer elektrischen Wicklung für eine direkte Messung meist nicht zugänglich ist, wird bei einem anderen Verfahren die mittlere Wicklungstemperatur vorteilhaft durch Widerstandsmessungen und programmierte Berechnungen, aufgrund der physikalischen Zusammenhänge, ermittelt. Die Wicklung selbst dient dabei als „Meßfühler" und übernimmt die Funktion eines Widerstandsthermometers.

Bekanntlich ist der Widerstand einer Wicklung außer vom Leiterwerkstoff, der Drahtstärke und der Windungszahl auch noch von der Temperatur $t$ abhängig. Die nichtlineare, jedoch gesetzmäßige Änderung des spezifischen Widerstandes wird der sogenannten Wärmebewegung der Atome und Moleküle, wie sie in jedem Stoff vorhanden sind, zugeschrieben. Da die Intensität dieser Bewegung mit steigender Temperatur zunimmt, bedeutet dies bei den meisten Metallen und Legierungen eine Erschwerung des Durchgangs freier Elektronen, resultierend also eine Erhöhung des Widerstandes.

Die Temperaturabhängigkeit des spezifischen Widerstandes – im vorliegenden Fall der Meßeffekt – kann in begrenzten Bereichen durch eine **Taylor-Reihe**

**Taylor-Reihe:** Grundform einer *Potenzreihe* zur numerisch verwertbaren Darstellung von Funktionen $y = f(x)$.

beschrieben werden. Soll ein Temperaturintervall von etwa −50 °C bis unterhalb des Schmelzpunktes erfaßt werden, und berücksichtigt man die werkstoffbedingten Streuungen der Koeffizienten, so ist es praktisch ausreichend, wenn man sich auf die ersten beiden Glieder der Potenzreihe mit dem linearen ($\alpha$) und dem quadratischen ($\beta$) Temperaturkoeffizienten beschränkt. Bei der Angabe von $\alpha$ bzw. $\beta$ ist zu beachten, daß der Wert des spezifischen Widerstandes genannt wird, da dieser sowohl von der Bezugstemperatur $t$ als auch von der Werkstoffzusammensetzung bzw. dem Reinheitsgrad des Metalls abhängt.

**Tabelle 1** Mittlere Werte der spezifischen Widerstände $\rho_{20}$ und der Temperaturkoeffizienten $\alpha_{20}$ und $\beta_{20}$ einiger Werkstoffe bei + 20 °C.

| Werkstoff | $\rho_{20} \cdot 10^{-3}$ [$\Omega \cdot mm^2/m$] | $\alpha_{20} \cdot 10^{-4}/K$ | $\beta_{20} \cdot 10^{-7}/K^2$ |
|---|---|---|---|
| Kupfer | 17,5 | 39,3 | 6 |
| Aluminium | 28,6 | 37,7 | 13 |
| Silber | 17 | 41 | 7 |
| Wolfram | 55 | 48 | 10 |
| Nickel | 85,5 | 47 | 90 |
| Platin | 108 | 39 | 6 |

## 2 Meßmittel und Daten

Die Raum- bzw. Objekttemperatur $t_k$ (°C) wird üblicherweise durch ein gutes Quecksilber- oder Digitalthermometer bestimmt. Zur Messung von Kaltwiderstand $R_k$ (Ohm) bzw. Warmwiderstand $R_w$ (Ohm) kann eine *Wheatstone-Brücke* oder ein *Digital-Ohmmeter**) mit geringen Fehlergrenzen verwendet werden. Sind die zu messenden Widerstände so niederohmig, daß Zuleitungs- und Übergangswiderstände nicht mehr zu vernachlässigen sind, so ist eine *Thomson-Brücke* oder ein geeignetes Digitalgerät in Vierpolschaltung*) (*Kelvin*) zu benutzen. Bei der Durchführung der $R$-Messungen sei noch auf zwei Punkte hingewiesen:

a) Der verwendete Meßgleichstrom soll möglichst klein sein, so daß die Eigenerwärmung der Wicklung keine wesentliche Verfälschung der Werte bewirkt. Der Einfluß ist im allgemeinen vernachlässigbar, wenn der Meßstrom $\leqslant 1\,\%$ vom Betriebsstrom gewählt wird.

b) Bei geringer Wärmekapazität des Prüflings, und/oder bei größeren Temperaturintervallen, ist während der Messung des Warmwiderstandes – die möglichst schnell erfolgen soll – darauf zu achten, daß dessen Änderung ein vertretbares Maß nicht überschreitet. Ist dies trotzdem der Fall, so kann der $R_w$-Wert – jedoch meist ungenauer – auch über die Betriebsdaten anhand vom Ersatzschaltbild ermittelt werden.

## 3 Programmierung mit TI-59/PC-100 C

Die theoretischen Zusammenhänge – in diesem Beitrag nicht dargestellt – sind zwar anschaulich, vom formelmäßigen Umfang her jedoch numerisch nicht gut zu handhaben. Für die praktische Anwendung wurden deshalb Programme aufgestellt, deren Befehle und Daten auf einer Magnetkarte untergebracht sind. Das Problem der Berechnung der mittleren Wicklungstemperatur $t_w$ oder des Warmwiderstandes $R_w$ reduziert sich dadurch auf das Einlesen der Karte, Auswahl des Programmparts „A", „B" oder „C" und Eingabe der gemessenen Daten. Innerhalb von 30 s bis 40 s liegt dann das gewünschte Ergebnis bereits vor.

Stellvertretend für die am Markt befindlichen Geräte wurde dabei der Taschenrechner TI-59 mit Thermodrucker PC-100 C von Texas Instruments verwendet. Die Rechner-Druckerkombination ist heute schon zu einem Kaufpreis von weniger als 1000 Mark erhältlich. Die beachtlichen algebraischen Eigenschaften und die Funktionen-Vielfalt bei gleichzeitig geringen Fehlergrenzen machen den TI-59 selbst für Anwendungen geeignet, wie sie gewöhnlich nur von größeren und aufwendigeren Geräten erfüllt werden. Mit Befehlen z.B. zur **indirekten Adressierung, Dekrementierung, Inkrementierung,** zur **Datenmanipulation** und zum Ausführen **logischer Vergleiche** in

**Indirekte Adressierung:** Nennung der Adresse, unter der noch nicht „direkt" der Operand zu finden ist, sondern erst eine weitere Adresse, die dann aber den Operanden enthält.

**Dekrementierung:** Herunterzählen um eine Einheit.

**Inkrementierung:** Erhöhung um eine Einheit.

**Datenmanipulation:** Sammelbezeichnung für alle denkbaren Operationen, durch die Daten sortiert, geformt oder verändert werden.

**Logischer Vergleich:** Prüfung darauf, ob zwei oder mehrere Größen einander gleich sind oder welche kleiner oder größer als andere sind. Auch die Prüfung darauf, ob eine Aussage „wahr" oder „falsch" ist, gehört dazu.

*) z.B. *Multimeter-Typ:* R 2302 — *Hersteller:* Hartmann & Braun AG., Frankfurt/M.
8800 A, 8500 A — Fluke GmbH., Düsseldorf

Verbindung mit 6 *Unterprogrammebenen, Label-Technik* usw. ergeben sich sehr viele Möglichkeiten der Programmgestaltung. Besonders erwähnt sei noch die Flexibilität des frei verfügbaren Gesamtspeicherplatzes mit 960 Bytes.

Abhängig von den jeweiligen Anforderungen kann von Hand oder per Programm die Verteilung zwischen **Programmspeicherstellen** und **Datenregistern** (DR) verändert werden. Man kann z.B. mit 960 Bytes ohne DR, aber auch mit 160 Bytes und 100 DR, arbeiten. Dazwischen sind, gestuft mit 80 Bytes ≙ 10 DR, vielerlei Kombinationen möglich.

## 4 Aufzeichnung auf Magnetkarte

Bei der erstmaligen Eingabe der Befehle und Daten (s. "**Listing**" und Tabelle 2) kann man wie folgt verfahren:

a) TI-59, Speicherbereichsverteilung (SBV): 319.79 (8 OP 17) von Hand einstellen; später erfolgt dies per Programm!

b) Register 60 ... 79 laden. Dabei ist ersichtlich, daß die Register DR 66 ... 71 doppelt, d.h. mit 2x6 Ziffern belegt sind. Da vom Eingabeformat bekanntlich nur max. 10 Ziffern parallel verarbeitet werden, ist hierbei gesplittet einzugeben. Beispiel: DR 67 wird zunächst mit 352664 („RK=") über STO 67 geladen. Anschließend wird 0.322330 („OHM") durch SUM 67 dazu addiert, so daß letztlich der 12-stellige Code 352624.322330 in DR 67 steht. Sinngemäß verfährt man mit den anderen Registern.

c) Programmbefehle gemäß Listing eintippen. Die Befehle HIRn und DSZm (m ⩾ 10) ergeben sich durch Überschreiben.

d) Da der alphanumerisch codierte Zeichenvorrat vom PC-100 C keine kleinen bzw. griechischen Buchstaben beinhaltet, gelten für den Schreibstreifen folgende Abkürzungen und Bezeichnungen:

A ≙ $\alpha$, B ≙ $\beta$: Temperaturkoeffizienten
TK ≙ $t_k$: „Kalt"-Temperatur (GRC: Grad Celsius)
RK ≙ $R_k$: „Kalt"-Widerstand (OHM)
TW ≙ $t_w$: „Warm"-Temperatur (GRC: Grad Celsius)
RW ≙ $R_w$: „Warm"-Widerstand (OHM)
K ≙ Grad Kelvin

e) Programmparts „A", „B" und „C" mit den angegebenen Beispielen auf Richtigkeit überprüfen.

f) SBV: 479.59 (6 OP 17) von Hand einstellen. Pgm. und Daten auf Block 1 und 2 der Magnetkarte aufzeichnen (1 bzw. 2 write). TI-59: OFF, ON. Karte einlesen und nochmaligen Test durchführen.

## 5 Benutzerhinweise und Beispiele

Kartenseiten 1 und 2 in der normalen SBV (479.59) einlesen.

| Taste | Werkstoff | Eingabe m. R/S | Berechnung |
|---|---|---|---|
| „A" | Kupfer | TK, RK, RW | $TW_{Cu}$ (°C) |
| „B" | wählbar | TK, RK, RW, A, B | TW (°C) |
| „C" | wählbar | A, B, TK, RK, TW | RW (Ω) |

Die Rechen- und Druckzeit pro Ablauf beträgt ca. 10 bis 15 s. Die Programmsegmente „A", „B" oder „C" können behelfsweise auch ohne Drucker verwendet werden, da die Daten bei jedem Durchlauf abgespeichert sind. TW ist bei Ablauf „A" und „B" in Register 5 bzw. RW bei Ablauf „C" in R 10 verfügbar.

Eine Fehleingabe der Temperaturkoeffizienten A u. od. B = 0 bewirkt im Ablauf „B" automatisch einen Programm-Stop (blinkende „0" im **Display**)!

Bei Ablauf „A" (Kupfer) erübrigt sich die Eingabe der Temperaturkoeffizienten (A, B), da deren Werte (s. Tabelle 1) zusammen mit den abgeleiteten Konstanten in R76 ... R79 enthalten sind.

*Beispiel zum Ablauf „A"*

Bei einer Raum- bzw. Objekttemperatur von TK = + 24 °C wurde an der Primärwicklung eines Netz-

**Programmspeicher:** Bei Taschenrechnern der Speicherbereich, der zur Aufnahme der Programmbefehle vorgesehen ist. Bei Mikrocomputern werden häufig die *Festwertspeicher* (ROM) so bezeichnet.

**Datenregister:** Bei Taschenrechnern der Speicherbereich, der zur Aufnahme variabler Größen (Daten) vorgesehen ist. Beim TI-59 ist der gesamt verfügbare Speicherbereich frei in beide Klassen aufteilbar („Speicherverteilung"). Bei Mikrocomputern werden häufig die *Schreib-Lesespeicher* (RAM) so bezeichnet.

**Listing:** Vollständiger Abdruck eines Programms, oft begleitet von Überschriften und Kommentaren.

**Display:** Bei Taschenrechnern wird die Leuchtdioden-, Fluoreszenz- oder Flüssigkristallanzeige so bezeichnet. Aber auch Bildschirme werden so genannt (CRT-Display).

transformators ein Gleichstrom-„Kalt"-Widerstand RK = 2,205 Ohm gemessen. Nach längerer Betriebszeit unter Vollast wurde unmittelbar nach dem Abschalten ein Gleichstrom-„Warm"-Widerstand von RW = 2,713 Ohm ermittelt. Die berechnete mittlere Wicklungstemperatur ist TW = + 92,84 °C (2-stellig gerundet).

| *Ausdruck:* | | *Belegung der Arbeitsregister:* | | |
|---|---|---|---|---|
| **„A":** KUPFER-WICKLUNG TEMP. BESTIMMUNG | | | | |
| TK= | | 80. | | 00 |
| 24. | GRC | 75. | | 01 |
| RK= | | 0. | | 02 |
| 2.205 | OHM | 9. | | 03 |
| RW= | | 4. | | 04 |
| 2.713 | OHM | 82.94197051 | TW | 05 |
| A = | | 24. | TK | 06 |
| 3.93-03 | /K | 2.205 | RK | 07 |
| B = | | 2.713 | RW | 08 |
| 6.-07 | /K² | 0. | | 09 |
| | | 0. | | 10 |
| TW= | | 0. | | 11 |
| 82.94 | GRC | -20. | | 12 |
| (GRAD CELSIUS) | | | | |
| xxxxxxxxxxxxxxxxxxxx | | | | |

*Beispiel zum Ablauf „B"*

An einer „60-Watt"-Glühlampe mit Wolfram-Wendel wurde bei Raumtemperatur von TK = + 21,5 °C ein Gleichstrom-„Kalt"-Widerstand RK = 52,83 Ohm gemessen.

Da eine direkte RW-Messung infolge der geringen Wärmekapazität schlecht möglich war, wurde der „Warm"-Widerstand aus den Betriebsdaten ermittelt. Bei einer Netzspannung von $U_N$ = 223,4 V wurde nach kurzer Zeit ein Strom von $I_S$ = 272,8 mA gemessen. Unter Vernachlässigung der induktiven Komponente ergibt sich somit für RW ≈ 818,91 Ohm. Für Wolfram gilt $A = 48 \cdot 10^{-4}/K$ u. $B = 10 \cdot 10^{-7}/K^2$. Die berechnete mittlere Temperatur der glühenden Wendel beträgt TW = + 2133,59 °C (2-stellig gerundet).

| *Ausdruck:* | | *Belegung der Arbeitsregister:* | | |
|---|---|---|---|---|
| **„B":** TEMP. BESTIMMUNG | | | | |
| TK= | | 13. | | 00 |
| 21.5 | GRC | 75. | | 01 |
| RK= | | 0. | | 02 |
| 52.83 | OHM | 11. | | 03 |
| RW= | | 1.5 | | 04 |
| 818.91 | OHM | 2133.589791 | TW | 05 |
| A = | | 21.5 | TK | 06 |
| 4.8-03 | /K | 52.83 | RK | 07 |
| B = | | 818.91 | RW | 08 |
| 1.-06 | /K² | 0.0048 | | 09 |
| | | 0.000001 | | 10 |
| TW= | | 4760000. | | 11 |
| 2133.59 | GRC | 2380. | | 12 |
| (GRAD CELSIUS) | | | | |
| xxxxxxxxxxxxxxxxxxxx | | | | |

*Beispiel zum Ablauf „C"*

Bei Raumtemperatur von TK = + 26 °C wurde an einer Drossel mit Kupferwicklung ($A = 39{,}3 \cdot 10^{-4}/K$, $B = 6 \cdot 10^{-7}/K^2$) ein Gleichstrom-„Kalt"-Widerstand von RK = 195,6 Ohm gemessen. Zur Beurteilung des Spannungsabfalles war bei einer Grenztemperatur von TW = + 115 °C der Gleichstrom-„Warm"-Widerstand zu bestimmen.

Die Berechnung ergibt für RW ≈ 263,468 Ohm

| *Ausdruck:* | | *Belegung der Arbeitsregister:* | | |
|---|---|---|---|---|
| **„C":** DATEN | | | | |
| | | 10. | | 00 |
| A = | | 17. | | 01 |
| 3.93-03 | /K | 0. | | 02 |
| B = | | 10. | | 03 |
| 6.-07 | /K² | 95. | | 04 |
| TK= | | 0.00393 | | 05 |
| 26. | GRC | 0.0000006 | | 06 |
| RK= | | 26. | TK | 07 |
| 195.6 | OHM | 195.6 | RK | 08 |
| TW= | | 115. | TW | 09 |
| 115. | GRC | 263.4681638 | RW | 10 |
| | | 130064.6326 | | 11 |
| | | 140064.6327 | | 12 |
| RW= | | 372664.2235 | | 13 |
| 263.4681638 | OHM | 352664.3223 | | 14 |
| | | 374364.2235 | | 15 |
| xxxxxxxxxxxxxxxxxxxx | | 354364.3223 | | 16 |

## 6 Programm-Listing

*Titel:* Mittlere Temperatur TW und „Warm"-Widerstand RW elektrischer Wicklungen.
Programm-Vorbereitung wahlweise mit Taste „A", „B" oder „C"; Daten eintasten und mit R/S eingeben.

| PSS | Code | Befehl | PSS | Code | Befehl | PSS | Code | Befehl | PSS | Code | Befehl | PSS | Code | Befehl |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 000 | 98 | ADV | 064 | 59 | 59 | 128 | 22 | INV | 192 | 95 | = | 256 | 98 | ADV |
| 001 | 06 | 6 | 065 | 00 | 0 | 129 | 86 | STF | 193 | 42 | STO | 257 | 08 | 8 |
| 002 | 35 | 1/X | 066 | 57 | 57 | 130 | 01 | 1 | 194 | 12 | 12 | 258 | 69 | OP |
| 003 | 82 | HIR | 067 | 69 | OP | 131 | 98 | ADV | 195 | 33 | X² | 259 | 17 | 17 |
| 004 | 05 | 5 | 068 | 05 | 05 | 132 | 25 | CLR | 196 | 75 | - | 260 | 69 | OP |
| 005 | 82 | HIR | 069 | 25 | CLR | 133 | 08 | 8 | 197 | 43 | RCL | 261 | 00 | 00 |
| 006 | 06 | 6 | 070 | 69 | OP | 134 | 69 | OP | 198 | 10 | 10 | 262 | 43 | RCL |
| 007 | 82 | HIR | 071 | 00 | 00 | 135 | 17 | 17 | 199 | 35 | 1/X | 263 | 75 | 75 |
| 008 | 07 | 7 | 072 | 92 | RTN | 136 | 06 | 6 | 200 | 95 | = | 264 | 69 | OP |
| 009 | 82 | HIR | 073 | 76 | LBL | 137 | 00 | 0 | 201 | 42 | STO | 265 | 01 | 01 |
| 010 | 08 | 8 | 074 | 19 | D' | 138 | 42 | STO | 202 | 11 | 11 | 266 | 69 | OP |
| 011 | 71 | SBR | 075 | 69 | OP | 139 | 01 | 01 | 203 | 09 | 9 | 267 | 05 | 05 |
| 012 | 00 | 0 | 076 | 00 | 00 | 140 | 87 | IFF | 204 | 42 | STO | 268 | 43 | RCL |
| 013 | 67 | 67 | 077 | 73 | RC* | 141 | 01 | 1 | 205 | 00 | 00 | 269 | 69 | 69 |
| 014 | 98 | ADV | 078 | 01 | 01 | 142 | 01 | 1 | 206 | 98 | ADV | 270 | 42 | STO |
| 015 | 91 | R/S | 079 | 69 | OP | 143 | 51 | 51 | 207 | 19 | D' | 271 | 11 | 11 |
| 016 | 76 | LBL | 080 | 01 | 01 | 144 | 03 | 3 | 208 | 43 | RCL | 272 | 43 | RCL |
| 017 | 15 | E | 081 | 71 | SBR | 145 | 44 | SUM | 209 | 06 | 06 | 273 | 70 | 70 |
| 018 | 98 | ADV | 082 | 00 | 0 | 146 | 01 | 01 | 210 | 75 | - | 274 | 42 | STO |
| 019 | 42 | STO | 083 | 67 | 67 | 147 | 05 | 5 | 211 | 02 | 2 | 275 | 12 | 12 |
| 020 | 03 | 03 | 084 | 73 | RC* | 148 | 61 | GTO | 212 | 00 | 0 | 276 | 43 | RCL |
| 021 | 19 | D' | 085 | 01 | 01 | 149 | 01 | 1 | 213 | 22 | INV | 277 | 66 | 66 |
| 022 | 25 | CLR | 086 | 22 | INV | 150 | 53 | 53 | 214 | 44 | SUM | 278 | 42 | STO |
| 023 | 91 | R/S | 087 | 59 | INT | 151 | 18 | C' | 215 | 12 | 12 | 279 | 13 | 13 |
| 024 | 17 | B' | 088 | 52 | EE | 152 | 03 | 3 | 216 | 54 | ) | 280 | 43 | RCL |
| 025 | 72 | ST* | 089 | 06 | 6 | 153 | 42 | STO | 217 | 42 | STO | 281 | 67 | 67 |
| 026 | 03 | 03 | 090 | 22 | INV | 154 | 02 | 02 | 218 | 04 | 04 | 282 | 42 | STO |
| 027 | 69 | OP | 091 | 52 | EE | 155 | 18 | C' | 219 | 65 | × | 283 | 14 | 14 |
| 028 | 23 | 23 | 092 | 69 | OP | 156 | 06 | 6 | 220 | 16 | A' | 284 | 43 | RCL |
| 029 | 97 | DSZ | 093 | 04 | 04 | 157 | 15 | E | 221 | 85 | + | 285 | 71 | 71 |
| 030 | 02 | 2 | 094 | 92 | RTN | 158 | 22 | INV | 222 | 01 | 1 | 286 | 42 | STO |
| 031 | 00 | 0 | 095 | 76 | LBL | 159 | 87 | IFF | 223 | 54 | ) | 287 | 15 | 15 |
| 032 | 21 | 21 | 096 | 10 | E' | 160 | 01 | 1 | 224 | 55 | ÷ | 288 | 43 | RCL |
| 033 | 92 | RTN | 097 | 01 | 1 | 161 | 01 | 1 | 225 | 16 | A' | 289 | 68 | 68 |
| 034 | 76 | LBL | 098 | 85 | + | 162 | 80 | 80 | 226 | 85 | + | 290 | 42 | STO |
| 035 | 16 | A' | 099 | 43 | RCL | 163 | 19 | D' | 227 | 43 | RCL | 291 | 16 | 16 |
| 036 | 73 | RC* | 100 | 05 | 05 | 164 | 43 | RCL | 228 | 04 | 04 | 292 | 07 | 7 |
| 037 | 00 | 00 | 101 | 65 | × | 165 | 76 | 76 | 229 | 33 | X² | 293 | 42 | STO |
| 038 | 69 | OP | 102 | 53 | ( | 166 | 52 | EE | 230 | 54 | ) | 294 | 00 | 00 |
| 039 | 20 | 20 | 103 | 16 | A' | 167 | 17 | B' | 231 | 65 | × | 295 | 01 | 1 |
| 040 | 92 | RTN | 104 | 75 | - | 168 | 19 | D' | 232 | 43 | RCL | 296 | 01 | 1 |
| 041 | 76 | LBL | 105 | 02 | 2 | 169 | 43 | RCL | 233 | 08 | 08 | 297 | 42 | STO |
| 042 | 17 | B' | 106 | 00 | 0 | 170 | 77 | 77 | 234 | 55 | ÷ | 298 | 01 | 01 |
| 043 | 69 | OP | 107 | 54 | ) | 171 | 52 | EE | 235 | 43 | RCL | 299 | 05 | 5 |
| 044 | 06 | 06 | 108 | 42 | STO | 172 | 17 | B' | 236 | 07 | 07 | 300 | 42 | STO |
| 045 | 69 | OP | 109 | 04 | 04 | 173 | 07 | 7 | 237 | 85 | + | 301 | 02 | 02 |
| 046 | 21 | 21 | 110 | 85 | + | 174 | 06 | 6 | 238 | 16 | A' | 302 | 15 | E |
| 047 | 92 | RTN | 111 | 43 | RCL | 175 | 42 | STO | 239 | 95 | = | 303 | 98 | ADV |
| 048 | 76 | LBL | 112 | 06 | 06 | 176 | 00 | 00 | 240 | 34 | √X | 304 | 19 | D' |
| 049 | 18 | C' | 113 | 65 | × | 177 | 61 | GTO | 241 | 75 | - | 305 | 10 | E' |
| 050 | 03 | 3 | 114 | 43 | RCL | 178 | 02 | 2 | 242 | 16 | A' | 306 | 32 | X:T |
| 051 | 42 | STO | 115 | 04 | 04 | 179 | 06 | 06 | 243 | 95 | = | 307 | 69 | OP |
| 052 | 59 | 59 | 116 | 33 | X² | 180 | 29 | CP | 244 | 42 | STO | 308 | 20 | 20 |
| 053 | 69 | OP | 117 | 95 | = | 181 | 43 | RCL | 245 | 05 | 05 | 309 | 10 | E' |
| 054 | 00 | 00 | 118 | 92 | RTN | 182 | 09 | 09 | 246 | 58 | FIX | 310 | 65 | × |
| 055 | 22 | INV | 119 | 76 | LBL | 183 | 67 | EQ | 247 | 02 | 2 | 311 | 43 | RCL |
| 056 | 52 | EE | 120 | 11 | A | 184 | 67 | EQ | 248 | 17 | B' | 312 | 08 | 08 |
| 057 | 73 | RC* | 121 | 86 | STF | 185 | 55 | ÷ | 249 | 22 | INV | 313 | 55 | ÷ |
| 058 | 01 | 01 | 122 | 01 | 1 | 186 | 02 | 2 | 250 | 58 | FIX | 314 | 32 | X:T |
| 059 | 84 | OP* | 123 | 61 | GTO | 187 | 55 | ÷ | 251 | 18 | C' | 315 | 95 | = |
| 060 | 59 | 59 | 124 | 01 | 1 | 188 | 43 | RCL | 252 | 81 | RST | 316 | 42 | STO |
| 061 | 69 | OP | 125 | 31 | 31 | 189 | 10 | 10 | 253 | 76 | LBL | 317 | 10 | 10 |
| 062 | 21 | 21 | 126 | 76 | LBL | 190 | 67 | EQ | 254 | 13 | C | 318 | 17 | B' |
| 063 | 97 | DSZ | 127 | 12 | B | 191 | 67 | EQ | 255 | 25 | CLR | 319 | 81 | RST |

**Tabelle 2**
Inhalt und Belegung der Datenregister 60 ... 79

| Code | | Text | R. Nr. |
|---|---|---|---|
| 2627413122. | | KLUNG | 60 |
| 3520432415. | | R-WIC | 61 |
| 2641332117. | | KUPFE | 62 |
| 3030413122. | | MMUNG | 63 |
| 1417363724. | | BESTI | 64 |
| 3717303340. | | TEMP. | 65 |
| 372664. | 223515 | TK= GRC | 66 |
| 352664. | 322330 | RK= OHM | 67 |
| 354364. | 322330 | RW= OHM | 68 |
| 130064. | 632600 | A = /K | 69 |
| 140064. | 632670 | B = /K² | 70 |
| 374364. | 223515 | TW= GRC | 71 |
| 3624413656. | | SIUS) | 72 |
| 1600151727. | | D CEL | 73 |
| 55223513. | | (GRA | 74 |
| 1613371731. | | DATEN | 75 |
| Cu | 0.00393 | $\alpha_{20}$ | 76 |
| | 0.0000006 | $\beta_{20}$ | 77 |
| | 9058958.333 | | 78 |
| | 3255. | | 79 |

## 7 Zusammenfassung

Die Beispiele machen deutlich, daß bei gleichzeitiger Verringerung der Fehlergrenzen und Verbesserung der numerischen Sicherheit, ein programmierbarer „Taschenrechner" vom Typ TI-59 einschließlich Drucker die Arbeiten ganz beträchtlich erleichtern kann.

Sind die Formeln, Gleichungen, Bedingungen etc. zur Lösung einer Aufgabe festgelegt, so können diese im „Lern"-Modus (LRN) direkt in den Programm- und Datenspeicher des TI-59 eingegeben und in der Betriebsart „Ablauf" (RUN) sofort verarbeitet werden. Die Programmspeicherstellen (PSS) können einzeln aufgerufen und die zugehörigen Befehle zur Kontrolle bzw. Änderung mit dem PC-100 C im Symboltext ausgedruckt werden. Bedingt durch die Benutzerfreundlichkeit des TI-59 (AOS-Eingabelogik usw.) ist ein „Programm" deshalb meist nicht mehr als die Folge der Tastenbefehle, die man bei der manuellen Lösung eines Problems vom Tastenfeld aus hätte eingeben müssen. Bei einfachen Rechenvorgängen und ein wenig Übung kann man davon ausgehen, daß ein „Programm" sich schon lohnt, wenn derselbe Algorithmus etwa 3 bis 5 mal benötigt wird.

VIEWEG

# Mathematicals in Hülle und Fülle

Martin Gardner
**Mathematisches Labyrinth**
Neue Probleme für die Knobelgemeinde
(Martin Gardner's Sixth Book of Mathematical Games from „Scientific American", dt.) (Aus dem Engl. übers. von R. Heersing und B. Kunisch). 1979. VI, 255 S. mit 190 Abb. Kart. 29,80 DM.
Inhalt: 50 mathematische Probleme von „Vier ungewöhnlichen Spielen" bis zu „Mathematischen Zaubertricks".
Der neue „Gardner" bringt wieder viele interessante Denkaufgaben. Sie sind spannend geschrieben und regen zum Nachdenken an. Ein Teil der Probleme sind Spiele und Spielereien. Hinter allen Rätseln und Problemen steht das Bemühen, den Leser in verschiedene Gebiete der Mathematik und Logik einzuführen, die keineswegs trivial sind.

Martin Gardner
**Logik unterm Galgen**
Ein Mathematical in 20 Problemen. VIII, 227 S. mit 125 Abb. Kart. 24,80 DM.

**Mathematische Knobeleien**
VIII, 204 S. mit 128 Abb. Gbd. 27,80 DM.

**Mathematische Rätsel und Probleme**
VII, 158 S. mit 89 Abb. Pb. 19,80 DM.

C. Stanley Ogilvy
**Mathematische Leckerbissen**
Über 150 noch ungelöste Probleme. 110 S. mit 39 Abb. Pb. 16,80 DM.

**Unterhaltsame Geometrie**
2., durchgesehene Auflage. 1979. 110 S. mit 132 Abb. Kart. 17,80 DM

M. Odier und Y. Roussel
**Trioker — mathematisch gespielt**
Logik und Fantasie mit Dreiecken.
(Surprenants Triangles No. 1, dt.) (Aus dem Franz. übers. von Barbara Friedl). 1979. 204 S. mit über 400 Abb. Kart. 24,80 DM.
Idee und Logik eines geometrischen Spiels mit 24 Dreiecksteinen werden hier vorgestellt.

Patrick Hughes und George Brecht
**Die Scheinwelt des Paradoxons**
Eine kommentierte Anthologie in Wort und Bild. 1978. VII, 120 S. mit 26 Abb. Gbd. 19,80 DM.
Dieses Buch lehrt nichts, führt zu nichts, klärt nichts, beabsichtigt nichts und ist deshalb eines der wenigen wichtigen Bücher.

**Fragen Sie Ihren Buchhändler nach diesen Büchern oder benutzen Sie die am Schluß eingeheftete Bestellkarte.**

In der Digitalelektronik – vorrangig bei Arbeiten mit Mikro- und Minicomputern und bei Logikanalysen – ist es vielfach notwendig, auf den verschiedenen Verarbeitungsebenen Konvertierungen zwischen den Zahlensystemen vorzunehmen. Problemlos und rationell, insbesondere für mehrstellige reelle positive Zahlen, ist dies mit einem programmierbaren Taschenrechner möglich. Dafür besonders gut geeignet ist der mittlerweile weitgestreute TI-59 einschließlich Thermodrucker PC-100 C von Texas Instruments.

W.-R. Haberditz

# Basis-Umwandlung von Zahlen mit dem TI-59

## 1 Anwendung und Beschreibung

Das lediglich auf einer Magnetkarte konservierte Programm basiert auf bekannten Umcodierungsformalismen – stellenrichtige Potenz-Division bzw. Addition etc. – und belegt 236 Programmspeicherstellen (PSS).

Es können folgende, in der numerischen Informatik interessierenden Umwandlungen damit durchgeführt werden:

„A": Dezimalzahl in den **Hexadezimal-Code;**
$\mathbf{DZ}_{x\,(10)} \rightarrow \mathbf{HEX}_{x\,(16)}$

„B": Dezimalzahl in den **Binär-Code;**
$\mathbf{DZ}_{x\,(10)} \rightarrow \mathbf{BN}_{x\,(2)}$

„C": Binär-Code in eine Dezimalzahl;
$\mathbf{BN}_{x\,(2)} \rightarrow \mathbf{DZ}_{x\,(10)}$

Die praktische Anwendung ist denkbar einfach. Nach dem Einlesen der beiden Kartenseiten – normale Einschalt-Speicherbereichsverteilung (SBV: 479.59) vom TI-59 – wird das Programm anhand der gewählten Umwandlungsart durch Drücken der Taste **„A", „B"** oder **„C"** vorbereitet. Dabei wird u.a. auch die im Ablauf vorgesehene SBV: 239.89 (9 OP 17) per „Pgm." eingestellt. Die Zahl bzw. Zahlenfolge wird jeweils mit „R/S" eingegeben. Abhängig vom Betrag bzw. der Stellenzahl ergeben sich für die gesuchte Lösung Rechen- und Druckzeiten von rund 15 ... 50 s. Rechnerintern arbeitet der TI-59 bekanntlich mit max. 13 CPU-Stellen. Zur Vermeidung von nichtmerkbaren Fehlrechnungen wurde deshalb der Bereich, beim Binär-Code zusätzlich auch noch der Stellenwert, per Pgm. begrenzt. Die Eingabe von Dezimalzahlen (parallel) mit $N \geqslant 1\,000\,000\,000\,016$ bzw. bei Binärzahlen mit $NBS \geqslant 41$ der seriell eingegebenen Stellen und deren Wert $\geqslant 2$ bewirken automatisch einen Programm-Stop (blinkendes Display!). In diesem Fall ist die Pgm.-Vorbereitung über die entsprechende Taste („A", „B" oder „C") zu wiederholen. Negative und/oder Zahlen mit Kommastellen (– N; N, xxx; – N, xxx) werden vom Pgm. als ganze Zahlen (|N| **Integer**) interpretiert, als solche ausgedruckt und verarbeitet. Die auf logarithmischer Basis arbeitende $y^x$-Funktion (**Approximation**) wurde in „A", „B" und „C" bewußt *nicht* angewandt, um absolut genaue numerische Lösungen zu erzielen. Mit den erwähnten Maßnahmen ist letztlich sichergestellt, daß die ausgedruckten Ergebnisse – bei größeren Zahlen nicht gut kontrollierbar – auch tatsächlich richtig sind! Zwecks besserer Übersicht für den Anwender sind auf dem Schreibstreifen die Ein- und Ausgabedaten mit entsprechendem Text gekennzeichnet (s. Beispiele).

Bedingt durch die Schreibbreite des PC-100 C mit max. 20 Druckpositionen ist bei Ablauf „A" und „B" ein serieller Ausdruck für die umcodierte Symbolfolge – Hexa 0 ... 9, A ... F bzw. Binär in 4er-Gruppen mit **0** und/oder **L** (z.B. L0LL) – erforderlich. Parallel geschrieben stellt dies dann jeweils die gesuchte Lösung dar.

Der alphanumerisch codierte Text sowie die Symboldaten sind in den Datenregistern 60 ... 89 gespeichert, wobei DR 74 ... 89 (s. Belegung) doppelt benutzt werden. Beispiel: Inhalt DR 84 = 13.27012701; die 13 ≙ A im Hexadezimal-Code bzw. 27012701 ≙ L0L0 im Binär-Code, wobei A bzw. L0L0 wiederum dem Wert 10 im Dezimalsystem entsprechen.

## 2 Aufzeichnung auf Magnetkarte

Bei der erstmaligen Eingabe der Befehle und Daten kann wie folgt verfahren werden:

a) TI-59, Speicherbereichsverteilung (SBV): 239.89 (9 OP 17) von Hand einstellen; später erfolgt dies per Programm! (s. PSS 017 ... 019)

b) Datenregister 60 ... 89 laden (Code siehe **Tabelle**)

c) Befehle gemäß Programm-Listing eintippen. DSZ 55 034 (PSS 041 ... 044) ergibt sich dabei durch die Tastenfolge DSZ, ÷, 0, $\sqrt{x}$ bzw. DSZ 57 200 (PSS 207 ... 210) durch DSZ, Eng, 2, 0

d) Programmsegmente mit den angegebenen Testbeispielen im Ablauf „A", „B" und „C" überprüfen!

e) SBV: 479.59 (6 OP 17) von Hand einstellen. Pgm. und Daten (Block 1 und 2) auf Magnetkarte aufzeichnen (1 bzw. 2 write)

f) TI-59: OFF, ON; Kartenseiten 1 und 2 einlesen und nochmaligen Test durchführen.

## 3 Testbeispiele und Hinweise

Kartenseiten 1 und 2 einlesen; SBV: 479.59

### 3.1 „UMWANDLUNG" einer „DEZIMAL-ZAHL X (10)" in den „HEXADEZIMAL-CODE"

Programm-Vorbereitung: Taste „A".

$DZ_{10}$ ins Anzeigeregister (AR) eintasten und Pgm. mit „R/S" starten. Ergebnis: 1 ... 9, A ... F; neue Berechnung, weiter mit „A".

*Beispiel 1:*

$DZ_{10}$ = 180150001 ≙ $HEX_{16}$ = ABCDEF1

Ausdruck:

„A": UMWANDLUNG

```
DEZIMAL-ZAHL:X(10)
  180150001.

HEXADEZIMAL-CODE:
          A
          B
          C
          D
          E
          F
          1
```

Rechen- und Druckzeit ca. 30 Sekunden

*Beispiel 2:*

Bedingt durch das max. 10-stellige Eingabeformat des TI-59 (bei E-Format max. 8-stellige Mantisse) müssen ganze Zahlen, mit mehr als 10 Stellen, gesplittet eingegeben werden.

$DZ_{10}$ = 1 000 000 000 001

1 EE 12 eintasten + 1; durch Drücken der „="-Taste steht jetzt die gewünschte Zahl im AR und das Pgm. kann wiederum mittels „R/S" gestartet werden.

$HEX_{16}$ = E8D4A51001

Ausdruck:

„A": UMWANDLUNG

```
DEZIMAL-ZAHL:X(10)
          1. 12   +
          1. 00   =
          1. 12

          1. 12

HEXADEZIMAL-CODE:
          E
          8
          D
          4
          A
          5
          1
          0
          0
          1
```

Rechen- und Druckzeit ca. 43 Sekunden

**Hexadezimal-Code:** Darstellung von Zahlen mit den Stellenwerten $16^0$, $16^1$, $16^2$ ... und den 16 Symbolen 0 bis 9, A bis F. Wegen des Zusammenhangs $2^4 = 16$ eignet sich dieser „Code" besonders als „Kurzschrift" für den Binärcode, weil je vier Binärstellen durch eine Hexadezimalstelle dargestellt werden können.

**Binär-Code:** Damit bezeichnet man das für maschinelle Verwendung besonders geeignete „Zweiersystem" mit den Stellenwerten $2^0$, $2^1$, $2^2$ ... und den zwei Ziffern 0 und 1. Für eine exakte Bezeichnungsweise kann folgende Festlegung dienen: Man benutzt „dual", wenn man das *Zahlensystem* und die Arithmetik meint. Die Bezeichnung „binär" dagegen ist dann richtig, wenn nur die zwei unterscheidbaren *Zustände* gemeint sind.

**Integer:** Fachausdruck für „ganze Zahl".

**Approximation:** Annäherung, Näherungsverfahren, Methode zur näherungsweisen Berechnung.

**0 oder L:** Bisher oft verwendete Bezeichnung für die beiden *Binärzustände.* Heute hat sich jedoch folgende Vereinbarung durchgesetzt: Die Elemente des dualen Zahlensystems werden mit 0 und 1 benannt; in technischen Darstellungen werden die beiden *Binärzustände* mit L (von *Low*) und H (von *High*) bezeichnet.

### 3.2 „UMWANDLUNG" einer „DEZIMAL-ZAHL X (10)" in den „BINAERCODE: X (2)"

Programm-Vorbereitung: Taste „B".

$DZ_{10}$ ins Anzeigeregister (AR) eintasten und Pgm. mit „R/S" starten. Ergebnis: 0 und/oder L in 4er-Gruppen; neue Berechnung, weiter mit „B"

*Beispiel 3:*

$DZ_{10}$ = 4275878553 ≙

$BN_2$ = LLLLLLLOLLOLLLOOLOLLLOLOLOOLLOOL

Ausdruck:

„B": UMWANDLUNG

```
DEZIMAL-ZAHL:X(10)
 4275878553.

BINAERCODE:X(2)
      LLLL
      LLLO
      LLOL
      LLOO
      LOLL
      LOLO
      LOOL
      LOOL
```

Rechen- und Druckzeit ca. 35 Sekunden

*Beispiel 4:*

$DZ_{10}$ = 1 000 000 000 001, Eingabe siehe Beispiel 2

$BN_2$ = LLLOLOOOLLOLOLOOLOLOOLOLOOOLOOOOOOOOOOL

Ausdruck:

„B": UMWANDLUNG

```
DEZIMAL-ZAHL:X(10)
          1. 12    +
          1. 00    =
          1. 12

          1. 12

BINAERCODE:X(2)
      LLLO
      LOOO
      LLOL
      OLOO
      LOLO
      OLOL
      OOOL
      OOOO
      OOOO
      OOOL
```

Rechen- und Druckzeit ca. 43 Sekunden

### 3.3 „UMWANDLUNG" vom „BINAERCODE X (2)" in eine „DEZIMAL-ZAHL: X (10)"

Programm-Vorbereitung: Taste „C".

NBS = Anzahl der Binärstellen ins Anzeigeregister (AR) eintasten und mit „R/S" eingeben. Binärzahl links beginnend seriell, d.h. stellenweise (0 oder 1), jeweils ins AR eintippen und mit „R/S" in die Register laden.

Ergebnis: $DZ_{10}$ (parallel); neue Berechnung, weiter mit „C"

*Beispiel 5:*

$BN_2$ = 100000001 ≙ $DZ_{10}$ = 257

NBS = 9

Ausdruck:

„C": UMWANDLUNG

```
NBS =
          9.

BINAERCODE:X(2)
          1.
          0.
          0.
          0.
          0.
          0.
          0.
          0.
          1.

DEZIMAL-ZAHL:X(10)
        257.
```

Rechen- und Druckzeit ca. 15 Sekunden

*Beispiel 6:*

$BN_2$ = 10101010101010101 ≙ $DZ_{10}$ = 87381

NBS = 17

Ausdruck:

„C": UMWANDLUNG

```
NBS =
         17.

BINAERCODE:X(2)
          1.
          0.
          1.
          0.
          1.
          0.
          1.
          0.
          1.
          0.
          1.
          0.
          1.
          0.
          1.
          0.
          1.

DEZIMAL-ZAHL:X(10)
      87381.
```

Rechen- und Druckzeit ca. 20 Sekunden

## 4 Programm-Listing

Titel: *Basis-Umwandlung von Zahlen*

| PSS | Code | Befehl | PSS | Code | Befehl | PSS | Code | Befehl | PSS | Code | Befehl | PSS | Code | Befehl |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 000 | 98 | ADV | 048 | 19 | D' | 096 | 22 | INV | 144 | 73 | RC* | 192 | 57 | 57 |
| 001 | 98 | ADV | 049 | 69 | OP | 097 | 87 | IFF | 145 | 02 | 02 | 193 | 04 | 4 |
| 002 | 91 | R/S | 050 | 05 | 05 | 098 | 01 | 1 | 146 | 87 | IFF | 194 | 44 | SUM |
| 003 | 76 | LBL | 051 | 69 | OP | 099 | 01 | 1 | 147 | 01 | 1 | 195 | 56 | 56 |
| 004 | 16 | A' | 052 | 00 | 00 | 100 | 07 | 07 | 148 | 01 | 1 | 196 | 34 | √X |
| 005 | 91 | R/S | 053 | 25 | CLR | 101 | 06 | 6 | 149 | 54 | 54 | 197 | 32 | X:T |
| 006 | 50 | I×I | 054 | 92 | RTN | 102 | 09 | 9 | 150 | 22 | INV | 198 | 03 | 3 |
| 007 | 59 | INT | 055 | 76 | LBL | 103 | 42 | STO | 151 | 59 | INT | 199 | 18 | C' |
| 008 | 99 | PRT | 056 | 10 | E' | 104 | 56 | 56 | 152 | 52 | EE | 200 | 16 | A' |
| 009 | 92 | RTN | 057 | 55 | ÷ | 105 | 04 | 4 | 153 | 08 | 8 | 201 | 77 | GE |
| 010 | 76 | LBL | 058 | 04 | 4 | 106 | 93 | . | 154 | 69 | OP | 202 | 77 | GE |
| 011 | 17 | B' | 059 | 33 | X² | 107 | 03 | 3 | 155 | 02 | 02 | 203 | 69 | OP |
| 012 | 25 | CLR | 060 | 95 | = | 108 | 59 | INT | 156 | 19 | D' | 204 | 20 | 20 |
| 013 | 06 | 6 | 061 | 42 | STO | 109 | 18 | C' | 157 | 43 | RCL | 205 | 72 | ST* |
| 014 | 69 | OP | 062 | 05 | 05 | 110 | 43 | RCL | 158 | 06 | 06 | 206 | 00 | 00 |
| 015 | 17 | 17 | 063 | 92 | RTN | 111 | 03 | 03 | 159 | 65 | × | 207 | 97 | DSZ |
| 016 | 47 | CMS | 064 | 76 | LBL | 112 | 22 | INV | 160 | 43 | RCL | 208 | 57 | 57 |
| 017 | 09 | 9 | 065 | 11 | A | 113 | 77 | GE | 161 | 05 | 05 | 209 | 02 | 2 |
| 018 | 69 | OP | 066 | 86 | STF | 114 | 01 | 1 | 162 | 95 | = | 210 | 00 | 00 |
| 019 | 17 | 17 | 067 | 01 | 1 | 115 | 27 | 27 | 163 | 22 | INV | 211 | 06 | 6 |
| 020 | 06 | 6 | 068 | 61 | GTO | 116 | 32 | X:T | 164 | 44 | SUM | 212 | 02 | 2 |
| 021 | 00 | 0 | 069 | 00 | 0 | 117 | 65 | × | 165 | 04 | 04 | 213 | 42 | STO |
| 022 | 42 | STO | 070 | 76 | 76 | 118 | 04 | 4 | 166 | 43 | RCL | 214 | 56 | 56 |
| 023 | 56 | 56 | 071 | 76 | LBL | 119 | 33 | X² | 167 | 05 | 05 | 215 | 04 | 4 |
| 024 | 02 | 2 | 072 | 12 | B | 120 | 95 | = | 168 | 10 | E' | 216 | 18 | C' |
| 025 | 76 | LBL | 073 | 22 | INV | 121 | 32 | X:T | 169 | 43 | RCL | 217 | 73 | RC* |
| 026 | 18 | C' | 074 | 86 | STF | 122 | 69 | OP | 170 | 04 | 04 | 218 | 00 | 00 |
| 027 | 42 | STO | 075 | 01 | 1 | 123 | 21 | 21 | 171 | 97 | DSZ | 219 | 65 | × |
| 028 | 55 | 55 | 076 | 17 | B' | 124 | 61 | GTO | 172 | 01 | 1 | 220 | 43 | RCL |
| 029 | 98 | ADV | 077 | 04 | 4 | 125 | 01 | 1 | 173 | 01 | 1 | 221 | 59 | 59 |
| 030 | 69 | OP | 078 | 18 | C' | 126 | 10 | 10 | 174 | 32 | 32 | 222 | 95 | = |
| 031 | 00 | 00 | 079 | 04 | 4 | 127 | 32 | X:T | 175 | 81 | RST | 223 | 44 | SUM |
| 032 | 22 | INV | 080 | 33 | X² | 128 | 10 | E' | 176 | 76 | LBL | 224 | 58 | 58 |
| 033 | 52 | EE | 081 | 32 | X:T | 129 | 32 | X:T | 177 | 13 | C | 225 | 02 | 2 |
| 034 | 73 | RC* | 082 | 25 | CLR | 130 | 42 | STO | 178 | 17 | B' | 226 | 49 | PRD |
| 035 | 56 | 56 | 083 | 16 | A' | 131 | 04 | 04 | 179 | 98 | ADV | 227 | 59 | 59 |
| 036 | 84 | OP* | 084 | 42 | STO | 132 | 55 | ÷ | 180 | 04 | 4 | 228 | 97 | DSZ |
| 037 | 55 | 55 | 085 | 03 | 03 | 133 | 43 | RCL | 181 | 01 | 1 | 229 | 00 | 0 |
| 038 | 01 | 1 | 086 | 75 | - | 134 | 05 | 05 | 182 | 32 | X:T | 230 | 02 | 2 |
| 039 | 44 | SUM | 087 | 01 | 1 | 135 | 85 | + | 183 | 43 | RCL | 231 | 17 | 17 |
| 040 | 56 | 56 | 088 | 42 | STO | 136 | 59 | INT | 184 | 73 | 73 | 232 | 43 | RCL |
| 041 | 42 | STO | 089 | 01 | 01 | 137 | 42 | STO | 185 | 69 | OP | 233 | 58 | 58 |
| 042 | 59 | 59 | 090 | 52 | EE | 138 | 06 | 06 | 186 | 01 | 01 | 234 | 99 | PRT |
| 043 | 97 | DSZ | 091 | 01 | 1 | 139 | 07 | 7 | 187 | 19 | D' | 235 | 81 | RST |
| 044 | 55 | 55 | 092 | 02 | 2 | 140 | 04 | 4 | 188 | 16 | A' | 236 | 00 | 0 |
| 045 | 00 | 0 | 093 | 95 | = | 141 | 95 | = | 189 | 77 | GE | 237 | 00 | 0 |
| 046 | 34 | 34 | 094 | 77 | GE | 142 | 42 | STO | 190 | 77 | GE | 238 | 00 | 0 |
| 047 | 76 | LBL | 095 | 77 | GE | 143 | 02 | 02 | 191 | 42 | STO | 239 | 00 | 0 |

**Tabelle**

Inhalt und Belegung der Datenregister 60 ... 89

| Code | Text | R. Nr. |
|---|---|---|
| 1627413122. | DLUNG | 60 |
| 4130431331. | UMWAN | 61 |
| 201560000. | 10) | 62 |
| 2327624455. | HL:X( | 63 |
| 1327204613. | AL-ZA | 64 |
| 1617462430. | DEZIM | 65 |
| 6244550356. | :X(2) | 66 |
| 3515321617. | RCODE | 67 |
| 1424311317. | BINAE | 68 |
| 1762000000. | E: | 69 |
| 2720153216. | L-COD | 70 |
| 1746243013. | EZIMA | 71 |
| 2317441316. | HEXAD | 72 |
| 3114360064. | NBS = | 73 |

| Code | Symbole | | R. Nr. |
|---|---|---|---|
| 1.01010101 | 0 | 0000 | 74 |
| 2.01010127 | 1 | 000L | 75 |
| 3.01012701 | 2 | 00L0 | 76 |
| 4.01012727 | 3 | 00LL | 77 |
| 5.01270101 | 4 | 0L00 | 78 |
| 6.01270127 | 5 | 0L0L | 79 |
| 7.01272701 | 6 | 0LL0 | 80 |
| 10.01272727 | 7 | 0LLL | 81 |
| 11.27010101 | 8 | L000 | 82 |
| 12.27010127 | 9 | L00L | 83 |
| 13.27012701 | A | L0L0 | 84 |
| 14.27012727 | B | L0LL | 85 |
| 15.27270101 | C | LL00 | 86 |
| 16.27270127 | D | LL0L | 87 |
| 17.27272701 | E | LLL0 | 88 |
| 21.27272727 | F | LLLL | 89 |

Man kann mit Hilfe von Steuertabellen seine Einkommensteuer berechnen, man kann sich natürlich auch vom Finanzamt überraschen lassen. Wer Überraschungen aber nicht liebt oder – seinem Gewissen folgend – den Steuerbescheid überprüfen möchte und gleichzeitig die nicht unerheblichen Ausgaben für die jährlich geänderten Steuertabellen scheut, wird im folgenden Beitrag erfahren, welche Abhilfe hier durch einen programmierbaren Taschenrechner möglich ist.

H. Alt

# Tabellenfreie Berechnung der Einkommensteuer mit Hilfe des programmierbaren Taschenrechners

## 1 Vorbemerkung

Eine interessante Anwendung der programmierbaren Taschenrechner liegt auf dem Gebiet der Steuerberechnung. Die Priorität des *Leistungsfähigkeitsprinzips* gegenüber dem *Äquivalenzprinzip* führt zu der Erkenntnis, daß die Fähigkeit, zur Finanzierung öffentlicher Erfordernisse beizutragen, mit wachsendem Einkommen überproportional ansteigt. Diesem Sachverhalt versucht man in den einzelnen Ländern nach mehr oder weniger kompliziert gestalteten progressiven Einkommensteuertarifen gerecht zu werden. Die unterschiedliche Ausgestaltung der Steuertarife rührt daher, daß jede wie immer geartete *Steuerprogression* nur dem Prinzip nach wirtschaftstheoretisch gerechtfertigt werden kann, nicht auch in bezug auf ihr konkretes Maß oder ihrem Steigungsverlauf [1]. Der Einsatz des programmierbaren Taschenrechners zur Steuerberechnung macht die bisher üblichen und jährlich neu zu erstellenden umfangreichen Steuertabellen entbehrlich.

Steueränderungen erfordern bei Einhaltung der prinzipiellen Tarifstruktur lediglich Änderungen innerhalb eines permanenten Datensatzes, auf den ein allgemein formuliertes Programm zurückgreift [2]. Die Realisierung einer programmierten Einkommensteuerberechnung soll mit den geltenden Steuertarifen in der Bundesrepublik Deutschland und in Österreich aufgezeigt werden. Die Programme hierzu sind für die programmierbaren UPN-Rechner der Typen HP 67 oder 97 in [3] angegeben.

Nachfolgend werden die für eine Programmierung des Steuertarifs erforderlichen Bestimmungen der jeweiligen Steuergesetze hervorgehoben und die Struktur der Datensätze für die tariflichen Festwerte angegeben. Mit Hilfe der programmierten Tarife sind z.B. Tarifvergleiche, wie in **Bild 1** angegeben, leicht durchzuführen.

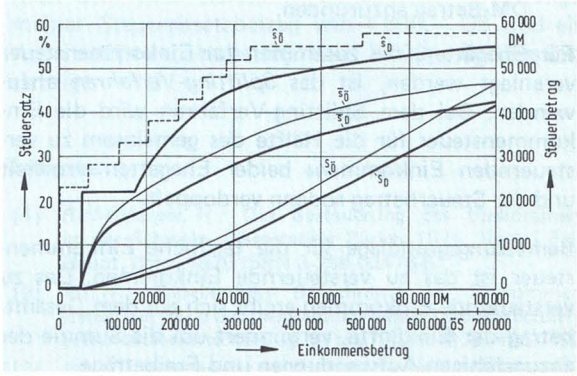



**Bild 1** Graphische Darstellung der Einkommensteuer als Ergebnis der programmierten Berechnung

## 2 Einkommensteuertarif gemäß § 32a ESTG, 1977, Bundesrepublik Deutschland

(1) Die tarifliche Einkommensteuer bemißt sich nach dem zu versteuernden Einkommen. Sie beträgt ab 1979 jeweils in Deutsche Mark

$$S^* = \begin{cases} 0 & x \leqslant 3\,719 \\ 0{,}22x - 812 & 3\,720 \leqslant x \leqslant 16\,019 \\ \{[10{,}86y - 154{,}42)\,y + 925]\,y + 2200\}\,y + 2708 & 16\,020 \leqslant y \leqslant 47\,999 \\ \{[0{,}1z - 6{,}07)\,z + 109{,}95]\,z + 4800\}\,z + 15\,298 & 48\,000 \leqslant z \leqslant 130\,019 \\ 0{,}56x - 13\,644 & 130\,020 \leqslant x \end{cases}$$

„$x$" ist das abgerundete zu versteuernde Einkommen. „$y$" ist ein Zehntausendstel des 16 000,– DM übersteigenden Teils des abgerundeten zu versteuernden Einkommens. „$z$" ist ein Zehntausendstel des 48 000,– DM übersteigenden Teils des abgerundeten zu versteuernden Einkommens.

(2) Das zu versteuernde Einkommen ist:

1. auf den nächsten durch 30 ohne Rest teilbaren vollen DM-Betrag abzurunden, wenn es nicht mehr als 48 000,– DM beträgt und nicht bereits durch 30 ohne Rest teilbar ist,
2. auf den nächsten durch 60 ohne Rest teilbaren vollen DM-Betrag abzurunden, wenn es mehr als 48 000,– DM beträgt und nicht bereits durch 60 ohne Rest teilbar ist.

(3) Die zur Berechnung der tariflichen Einkommensteuer erforderlichen Rechenschritte sind in der Reihenfolge auszuführen, die sich nach dem *Horner-Schema* ergibt. Dabei sind die sich aus den Multiplikationen ergebenden Zwischenergebnisse für jeden weiteren Rechenschritt mit drei Dezimalstellen anzusetzen; die nachfolgenden Dezimalstellen sind fortzulassen. Der sich ergebende Steuerbetrag ist auf den nächsten vollen DM-Betrag abzurunden.

Für Ehegatten, die zusammen zur Einkommensteuer veranlagt werden, ist das *Splitting-Verfahren* anzuwenden. Bei dem Splitting-Verfahren wird die Einkommensteuer für die Hälfte des gemeinsam zu versteuernden Einkommens beider Ehegatten ermittelt und der Steuerbetrag sodann verdoppelt.

Bemessungsgrundlage für die tarifliche Einkommensteuer ist das zu versteuernde Einkommen. Das zu versteuernde Einkommen ergibt sich aus dem Gesamtbetrag der Einkünfte, vermindert um die Summe der abzugsfähigen Aufwendungen und Freibeträge.

## 3 Einkommensteuertarif gemäß § 33 ESTG, 1972, Österreich

(1) Die Einkommensteuer von dem zu versteuernden Einkommen beträgt 1978 jährlich

für die ersten 50 000 S ... 23 v.H.
für die weiteren 50 000 S ... 28 v.H.
für die weiteren 50 000 S ... 33 v.H.
für die weiteren 50 000 S ... 38 v.H.
für die weiteren 40 000 S ... 43 v.H.
für die weiteren 40 000 S ... 48 v.H.
für die weiteren 40 000 S ... 52 v.H.
für die weiteren 180 000 S ... 55 v.H.
für die weiteren 500 000 S ... 58 v.H.
für die weiteren 500 000 S ... 60 v.H.
für alle weiteren Beträge ..... 62 v.H.

(2) Wenn das zu versteuernde Einkommen nicht durch 100 S teilbar ist, so sind Restbeträge bis einschließlich 50 S zu vernachlässigen und Restbeträge von mehr als 50 S als volle 100 S zu rechnen.

(3) Dem Steuerpflichtigen steht ein allgemeiner Steuerabsetzbetrag in Höhe von 4 400 S jährlich zu.

(4) Ein Alleinverdienerabsetzbetrag in Höhe von 2 400 S jährlich ...

(5) Ein Arbeitnehmerabsetzbetrag in Höhe von 2 000 S jährlich ...

⋮

(8) Die Einkommensteuer wird nicht erhoben, wenn sie den Betrag von 300 S nicht übersteigt. Übersteigt die Einkommensteuer den Betrag von 300 S, dann wird sie

bis zu einem Betrag von 350 S mit 150 S,
bis zu einem Betrag von 400 S mit 200 S und
bis zu einem Betrag von 450 S mit 300 S

erhoben.

Nach dem Stand vom 1. Januar 1979 [1] gelten für die Absetzbeträge ab 1979 folgende erhöhten Werte:

Allgemeiner Steuerabsetzbetrag (§ 57 Abs. 1) jährlich 4 800 S
Alleinverdienerabsetzbetrag (§ 57 Abs. 2) jährlich 3 200 S
Arbeitnehmerabsetzbetrag (§ 57 Abs. 3) jährlich 3 000 S.

In der Jahresausgleichstabelle sind neben den Absetzbeträgen der Werbungskostenpauschbetrag von 4 914 S und der Sonderausgabenpauschbetrag von 3 276 S berücksichtigt.

## 4 Permanente Daten

Die Tarifstruktur gemäß den erstellten Einkommensteuerprogrammen benötigt einen permanenten Datensatz. Für die beiden vorgenannten Tarife ergeben sich z.B. folgende Datensätze:

a) *Bundesrepublik Deutschland*

Primärspeicher

| | ab 1979 | 1978 | 1977 |
|---|---|---|---|
| STO 1: | 0,22 | 0,22 | 0,22 |
| STO 2: | 812 | 726 | 660 |
| STO 3: | 48000 | 48000 | 48000 |
| STO 4: | 16019 | 16019 | 16019 |
| STO 5: | 10,86 | 0 | 0 |
| STO 6: | −154,42 | −49,2 | −49,2 |
| STO 7: | 925 | 505,3 | 505,3 |
| STO 8: | 2200 | 3077 | 3077 |
| STO 9: | 2708 | 2792 | 2858 |

Sekundärspeicher

| | ab 1979 | 1978 | 1977 |
|---|---|---|---|
| STO 1′: | 0,56 | 0,56 | 0,56 |
| STO 2′: | 13644 | 12742 | 12676 |
| STO 3′: | 130019 | 130019 | 130019 |
| STO 4′: [1] | (5344) | (5344) | (4384) |
| STO 5′: | 0,1 | 0,1 | 0,1 |
| STO 6′: | −6,07 | −6,07 | −6,07 |
| STO 7′: | 109,95 | 109,95 | 109,95 |
| STO 8′: | 4800 | 4800 | 4800 |
| STO 9′: | 15298 | 16200 | 16266 |

b) *Österreich*

Primärspeicher

| | Jahr 1978 | 1979 |
|---|---|---|
| STO 4: | −4 400 | −4 800 |
| STO 5: | −2 000 | −3 000 |
| STO 6: | −2 400 | −3 200 |
| STO 7: | −4 914 | −4 914 |
| STO 8: | −3 276 | −3 276 |
| STO 9: | 0,62 | 0,62 |

Sekundärspeicher

| | |
|---|---|
| STO 0′: | 50 000,23 |
| STO 1′: | 50 000,28 |
| STO 2′: | 50 000,33 |
| STO 3′: | 50 000,38 |
| STO 4′: | 40 000,43 |
| STO 5′: | 40 000,48 |
| STO 6′: | 40 000,52 |
| STO 7′: | 180 000,55 |
| STO 8′: | 500 000,58 |
| STO 9′: | 500 000,60 |

[1] Diese Werte werden nur bei Berücksichtigung der Pauschalbeträge zur Ermittlung des zu versteuernden Einkommens benötigt (Programmvariante Clear Flag 1).

## 5 Ergebnisse

Als Ergebnisse der programmierten Berechnung können z.B. folgende Werte ausgegeben werden:

1. Spitzensteuersatz $\hat{s} = \left.\frac{dS}{dE}\right|_E$ in Prozent
2. Durchschnittssteuersatz $\bar{s} = \frac{S}{E}$ in Prozent
3. Steuerbetrag $S$ in der Landeswährung
4. verbleibender Einkommensbetrag $E_N = E - S$ in der Landeswährung.

Die Rechenzeit für einen Eingabewert liegt zwischen 10 und 30 Sekunden.

In dem angegebenen Diagramm **(Bild 1)** sind die Ergebnisse einer programmierten Einkommensteuerberechnung bezogen auf das zu versteuernde Einkommen für Ledige mit den Tarifdaten für 1979 dargestellt. Dabei wurde ein Umrechnungskurs zwischen DM und ÖS von 13,70 DM/100 ÖS als Mittelkurs zwischen Ankaufkurs und Verkaufkurs vom 7.3.1979 zugrunde gelegt.

Um vergleichbare Verhältnisse zu schaffen, ist bei den Kennlinien nach dem deutschen Steuerrecht ein Weihnachtsfreibetrag von 400,– DM, ein Arbeitnehmerfreibetrag von 480,– DM und ein Tariffreibetrag von 512,– DM berücksichtigt. Bei den Kennlinien nach dem österreichischen Steuerrecht ist ein allgemeiner Steuerabsetzbetrag von 4800,– ÖS und ein Arbeitnehmerabsetzbetrag von 3000,– ÖS berücksichtigt.

### Literatur

[1] *Richenberger, H.:* Die Besteuerung des Einkommens in der Schweiz, Dissertation Zürich 1975, Verlag Paul Haupt, Bern, 1976 ISBN 3-258-02476-6.

[2] *Alt, H.:* Tabellenfreie Berechnung der Einkommensteuer mit Hilfe des programmierbaren Taschenrechners, Elektrizitätswirtschaft Jg. 77(1978), H.14, S. 486–489.

[3] *Alt, H.:* Anwendung programmierbarer Taschenrechner, Band 1, 1979, Verlag Vieweg, Braunschweig/Wiesbaden, ISBN 3-528-04150-1.

Im folgenden soll das Eingabe- und Rechensystem „Umgekehrte Polnische Notation" (UPN) vorgestellt werden. Es wird also aufgezeigt, was UPN ist, wie man damit arbeitet und wo die unbestreitbaren Vorteile dieses Systems liegen. Gleichzeitig erfährt der Leser auch einiges über die bisherige und zukünftige Entwicklung im Bereich der elektronischen Taschen- und Kleinrechner.

Chr. Fuchs und K.-H. Gosmann

# UPN – Die zukunftsorientierte Computer-Logik

Im Jahre 1972 wurde der erste naturwissenschaftliche Taschenrechner der Welt, der HP-35, von der Firma *Hewlett-Packard* (HP) vorgestellt. Er arbeitete, wie alle nachfolgenden Taschen- und Kleinrechner dieses Herstellers, mit dem äußerst rationellen und wirkungsvollen System UPN. Nach jahrelangen Untersuchungen war dieses System als das bestgeeignete von HP angesehen worden (die Gründe dafür werden später erläutert) und hatte sich zudem im Tischcomputerbereich (HP-9100 A, HP-9100 B) bereits hervorragend bewährt.

Zunächst zur Nomenklatur: Mit der Bezeichnung „polnisch" nimmt diese Notation Bezug auf den polnischen Mathematiker *Jan Lukasiewicz,* der zur Darstellung von Aussagen der formalen Logik eine symbolische, eindeutige Schreibweise ohne Klammersetzungen entwickelte. „Umgekehrt" besagt darin, daß die mathematische Verknüpfung (Operation) zwischen Variablen erst im Nachhinein angegeben wird. Eine einfache Addition zweier Variablen $a$ und $b$ wird also gemäß der „nachgestellten" Schreibweise $a, b, +$ vollzogen, Im Gegensatz zu der „eingeschobenen" Schreibweise $a + b$ der Algebra, die dann zur Wahrung der Hierarchie mathematischer Operationen Klammersetzung erfordert.

Das UPN-Logiksystem wurde in Kombination mit einem *Stack-Register* von Hewlett-Packard für Taschenrechner entwickelt. Nach diesem System arbeitende Rechner sind schon äußerlich an der ENTER-Taste zu erkennen; dafür sucht man vergeblich nach Klammer-Tasten und der „Ist-gleich-Taste".

Einige wesentliche Vorteile des UPN-Logiksystems sind:

- Bei der Eingabe über das Tastenfeld geht man bei allen Berechnungen stets so vor, wie man das Problem auch mit anderen Hilfsmitteln (z.B. mit Bleistift und Papier) lösen würde. Problemstellungen müssen nicht erst für den Rechner umformuliert werden.
- Jeder Druck auf eine Rechen- oder Funktionstaste bewirkt (und zwar ohne Ausnahmen), daß die entsprechende Operation unmittelbar ausgeführt und das jeweilige Zwischenresultat angezeigt wird. Das fördert die Übersichtlichkeit und ermöglicht neben der laufenden Kontrolle der Zahlenwerte ein jederzeitiges Abweichen vom geplanten Rechengang.
- Während die Transparenz dieses Systems die Fehlererkennung erleichtert, stehen sehr effektive Korrektur-Funktionen zur Berichtigung solcher Fehler zur Verfügung. Bereits eingetastete aber noch nicht verarbeitete fehlerhafte Daten kann jeder Rechner leicht löschen; das HP-Logiksystem (UPN) mit seinem speziellen "LAST X"-Register ermöglicht dagegen auch die mühelose Korrektur, wenn fehlerhafte Daten schon in den Rechengang einbezogen wurden oder aber mit den richtigen Zahlenwerten eine falsche Rechnung ausgeführt wurde.

So vielseitig dieses Logiksystem ist, so einfach ist seine Handhabung. Die *Grundregel* lautet:

1. Die Zahl wird eingegeben.
2. Wenn eine mathematische Operation ausgeführt werden kann, ist die entsprechende Taste zu drücken. Andernfalls drückt man ENTER.

Nach dieser einfachen Regel verfährt man bis zum Erhalt des Endresultats. Das zugehörige Ablaufdiagramm ist in **Bild 1** dargestellt.

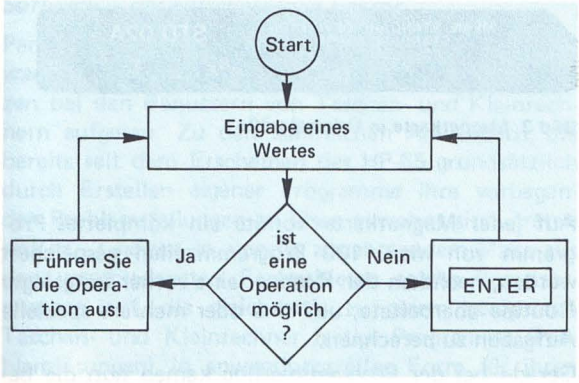



**Bild 1** Ablaufdiagramm für das Arbeiten im UPN-Logiksystem

Eine einfache Addition zweier Variablen *a* und *b* erfolgt also entsprechend Bild 1 folgendermaßen:

*Beispiel:* $a = 4{,}25$; $b = 3{,}75$

Eingabe in den Rechner:

ENTER 4,25.

Die ENTER-Taste dient zur Trennung der Variablen. Erfolgt kein Drücken der ENTER-Taste, wird $b = 3{,}75$ als 3., 4. und 5. Nachkommastelle vom Rechner gelesen. Die richtige Verwendung der ENTER-Taste und das weitere Eintasten von 3,75 + entspricht der nachgestellten Schreibweise *a, b,* +. Um erkennen zu können, welches System die Handhabung der UPN-Rechner so erleichert, wird der interne Ablauf im *Rechenregister-Stapel* (*Stack*) erläutert (**Bild 2**).

Das in Bild 2 angezeigte *X*-Register ist eines von insgesamt vier Speicherregistern, die im Innern des Rechners zu einem automatischen Register-Stapel (engl. *Stack*) zusammengefaßt sind. Die Stack-Register sind mit *X, Y, Z* und *T* bezeichnet und sind übereinandergestapelt.

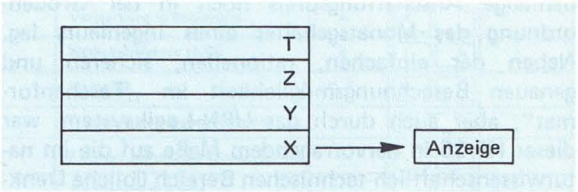



**Bild 2** Schematischer Aufbau des Stack-Registers mit den Einzelregistern X, Y, Z und T

Der Inhalt des *X*-Registers erscheint stets in der Anzeige. In diesem Register wird auch jede über die Tastatur eingegebene Zahl gespeichert. Durch Drücken von ENTER wird der Inhalt vom *X*- in das *Y*-Register kopiert, wobei zusätzlich der Ursprungswert zur optischen Kontrolle im *X*-Register erhalten bleibt. Eine jetzt im Anschluß an ENTER eingetastete Zahl überschreibt diesen alten Wert. Da jetzt beide erforderlichen Zahlen gespeichert sind, kann der Rechenschritt ausgeführt werden. Dazu drückt man lediglich die entsprechende Funktionstaste (+, –, x, :). Der Rechner kombiniert dann die beiden Ausgangswerte zum Resultat und verschiebt automatisch alle Stack-Register-Inhalte um eine Position nach unten. Auf diese Weise steht das Ergebnis des Rechenschrittes direkt in der Anzeige. Die Eleganz dieses Systems besteht darin, daß jetzt, wenn eine weitere Zahl eingegeben wird, die Inhalte des Stacks automatisch angehoben werden (*Stack-Lift*).

*Beispiel:*

$$\sqrt{5\,\{[(5{,}5-2{,}5)\times(6{,}8-1{,}3)^2]^{1/3}+1\}}$$

| | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| T | | | | | | | | | | | | | | | | | | |
| Z | | | | | | 3 | 3 | | | | | | | | | | | |
| Y | | 5,5 | 5,5 | | 3 | 6,8 | 6,8 | 3 | 3 | | 90,75 | 90,75 | | 4,49 | | 5,49 | | |
| X | 5,5 | 5,5 | 2,5 | 3 | 6,8 | 6,8 | 1,3 | 5,5 | 30,25 | 90,75 | 3 | 0,33 | 4,49 | 1 | 5,49 | 5 | 27,47 | 5,24 |
| Taste | 5,5 | ↑ | 2,5 | – | 6,8 | ↑ | 1,3 | – | $x^2$ | × | 3 | 1/x | $y^x$ | 1 | + | 5 | × | √ |

Wie aus diesem Beispiel zu ersehen ist, werden alle Zahlen automatisch innerhalb des Stack verschoben, und das Zwischenergebnis wird jeweils direkt angezeigt. Die Zwischenwerte müssen also nicht, wie normalerweise, in einzelne Datenregister abgespeichert werden. Außerdem entfällt das Eintasten der Klammerebenen, die Neuformatierung und vor allem die Gleichheitszeichentaste.

Dem im Jahre 1972 vorgestellten HP-35 war ein außerordentlicher Erfolg beschieden, obwohl der damalige Anschaffungspreis noch in der Größenordnung des Monatsgehaltes eines Ingenieurs lag. Neben der einfachen, rationellen, sicheren und genauen Berechnungsmöglichkeit im „Taschenformat", aber auch durch das UPN-Logiksystem, war dieser HP-35 in hervorragendem Maße auf die im naturwissenschaftlich-technischen Bereich übliche Denkweise zugeschnitten. Dazu trägt nicht nur bei, daß durch UPN mit dem Aufarbeiten mathematisch formulierter Probleme auch automatisch ein gedankliches Nachvollziehen der Lösung verbunden ist. Vielmehr ließen sich nun auch Konvergenz, Divergenz und asymptotisches Verhalten auf eine „neuartige" Weise veranschaulichen, was durch die sich daraus zwangsläufig ergebende Frage nach dem „warum ist das so" zu einem tiefen Verständnis des vorgegebenen Problems führt. Ebenso schnell und zuverlässig war nun das Abschätzen in Größenordnungen und – da durch die UPN zusammen mit dem Stack-Register auch alle Zwischenergebnisse direkt zur Verfügung stehen – das Erkennen von „wesentlichen" und „unwesentlichen" Beträgen zu den berechneten Endergebnissen möglich.

Zwangsläufig war damit die Ablösung des Rechenstabes – eines der wichtigsten Arbeitsgeräte von Wissenschaftlern, Ingenieuren und Technikern – verbunden. Ebenso wanderten zahlreiche Tabellenwerke der trigonometrischen, logarithmischen, Potenz-, Exponential- und hyperbolischen Funktionen endgültig vom Arbeitstisch in die Schublade. Langwieriges Aufrunden oder Interpolieren von Tabellenwerten gehörte ab jetzt der Vergangenheit an. Statt dessen: Zwei Tastendrucke, und das Endergebnis lag vor!

In den folgenden Jahren schlossen sich immer mehr Hersteller naturwissenschaftlich-technischer Taschenrechner dieser Entwicklung an. Entsprechend schnell nahm die Typenvielfalt in einem bis heute anhaltenden Trend zu. Doch schon im Jahre 1975 war es wiederum die Firma Hewlett-Packard, die, wie es in der damaligen Fachpresse hieß, einen entscheidenden Schritt in der Rechnertechnologie tat. Es wurde nämlich der erste voll programmierbare Taschenrechner der Welt, der HP-65, vorgestellt, der neben der freien Programmierbarkeit die Möglichkeit besaß, einmal erstellte Programme extern auf Magnetkarten zu speichern und auf Abruf in Sekundenschnelle über einen Magnetkartenleser wieder einzulesen. Aber anstatt, wie es bei den Computern üblich war, diese Programme auf großen Magnetspulen oder Lochkartenstapeln zu speichern, arbeitete der HP-65 mit kleinen Magnetkarten. In **Bild 3** ist die Originalgröße solch einer Magnetkarte gezeigt.

**Bild 3** Magnetkarte in Originalgröße

Auf jeder Magnetkarte konnte ein komplettes Programm von max. 100 Programmzeilen gespeichert werden, nachdem der Rechner eine vorher festgelegte Routine abarbeitete, um eine oder mehrere spezielle Aufgaben zu berechnen.

Gerade bei der Programmierung kamen nun die besonderen Vorzüge der UPN ein zweites Mal zum Tragen. Bei der begrenzten Speicherkapazität von Programmzeilen konnte nämlich das UPN-Logiksystem für eine Rationalisierung sorgen. Denn einmal erspart die Klammerfreiheit dieses Systems, insbesondere bei sehr verschachtelten Formeln, eine ganze Reihe von Klammersetzbefehlen, zum zweiten entfallen durch die Stack-Technik die gerade mit anderen Eingabesystemen häufig notwendigen *Transferbefehle* (*Store* und *Recall*) zum Zwischenspeichern und Wiederabrufen von Zwischenergebnissen.

Die so mit der Hardware den Rechnern vorgegebene Benutzerfreundlichkeit wird durch den hohen Qualitätsmaßstab ergänzt, den Hewlett-Packard an all seine Produkte anlegt. Zusätzlich steht dem Benutzer ein umfangreiches Software-Angebot aus den verschiedensten Fachbereichen zur Verfügung.

*Fazit:*

In den vergangenen Jahren waren grundsätzlich zwei verschiedene Trends erkennbar. Einerseits konnte durch die Miniaturisierung und die Großserienherstellung von Bauteilen der Elektronik die Leistungsfähigkeit der Rechner stark anwachsen; insbesondere deutlich sichtbar war dies in stetigem Anwachsen

der Größe von Programm- und Datenspeichern, festintegrierten Statistik-Registern, aber auch durch „ausgeklügelte" Programmiermöglichkeiten wie Speicherarithmetik, indirekte und relative Adressierung. Andererseits wurde als Folge der Miniaturisierung mit *Read Only Memories* (ROM) eine Verbesserung der Algorithmen erreicht, so daß die heutigen Taschen- und Kleinrechner einen derartig hohen technischen Standard besitzen, der vor 10 Jahren selbst von „ausgewachsenen" Tischcomputern bei mehr als 10fachem Anschaffungspreis nicht erreicht wurde.

### *Software – der entscheidende Faktor*

Parallel zu den Entwicklungen im Bereich der *Hardware* haben sich in jüngerer Zeit auch neue Tendenzen bei den Benutzern von Taschen- und Kleinrechnern aufgetan. Zu den zahlreichen Anwendern, die bereits seit dem Erscheinen des HP-65 grundsätzlich durch Erstellen eigener Programme ihre vorliegenden Problemstellungen zu lösen gewohnt sind, treten seit ca. 2 Jahren in ständig zunehmendem Maße aus den verschiedensten Fachgebieten solche Benutzergruppen auf, die gleichzeitig zu einem geeigneten Taschen- und Kleinrechner fertige Programme (Problemlösungen) in anwendungsreifer Form für ihren jeweiligen Arbeitsbereich wünschen. Die Firma Hewlett-Packard hat diesen Trend frühzeitig erkannt und im Taschen- und Kleinrechnerbereich maßgeblich beeinflußt.

Zur Zeit werden von Hewlett-Packard ca. 3000 Programme aus den verschiedensten naturwissenschaftlichen und finanzmathematischen Bereichen, z.B. Statistik, Mathematik, Elektrotechnik, Maschinenbau, Vermessung, Navigation etc., angeboten. Darüberhinaus stehen dem Benutzer von HP-Taschenrechnern weitere Programme, „vom Fachmann für den Fachmann" entwickelt, in einer anwenderbezogenen Form zur Verfügung. Diese Programme wurden durch eine enge Kooperation zwischen HP und qualifizierten Programmentwicklern erstellt, die neben hervorragenden Fachkenntnissen ebenso über exzellente Programmierkenntnisse verfügen.

Um die Leistungsfähigkeit, aber auch die Bedienungsfreundlichkeit und Handhabung der einzelnen Programme zu demonstrieren, soll der Eingabemodus „Entwurf aktiver Filter" dargestellt werden. Das Programm berechnet die Schaltelement-Werte für die angegebenen Standard-Filterschaltungen. Der Benutzer hat die Grenzfrequenz oder Mittenfrequenz $f_0$, die Verstärkung in Bandmitte ($A$), den Welligkeitsfaktor $a$ und eine Kapazität $C$ vorzugeben. Das Programm gibt daraufhin eine Liste der Schaltelemente aus, mit deren Hilfe sich das gewünschte Filter realisieren läßt. Die nachfolgende Tabelle zeigt die Anweisungsfolge.

| Nr. | Anweisung | Werte | Tasten | Anzeige |
|---|---|---|---|---|
| 1 | Programm einlesen | | | |
| 2 | Geben Sie die gewünschten | | | |
| | Filtereigenschaften ein | | | |
| | Grenz- oder Mittenfrequenz | $f_0$, Hz | f a | |
| | Verstärkung in Bandmitte | A | A | |
| | Welligkeitsfaktor (1/Q) | $\alpha$ | f b | |
| | Kapazität | C, F | B | |
| 3 | Wählen Sie die entsprechende | | | |
| | Filterart aus und berechnen Sie die | | | |
| | Schaltelement-Werte | | | |
| | Tiefpaß | | C | $R_1$, |
| | | | | $C_2$, |
| | | | | $R_3$, |
| | | | | $R_4$, |
| | | | | $C_5$, |
| | Hochpaß | | D | $C_1$, |
| | | | | $R_2$, |
| | | | | $C_3$, |
| | | | | $C_4$, |
| | | | | $R_5$, |
| | Bandpaß | | E | $R_1$, |
| | | | | $R_2$, |
| | | | | $C_3$, |
| | | | | $C_4$, |
| | | | | $R_5$ |

*Beispiel:*

Entwerfen Sie einen aktiven Hochpaß mit folgenden Eigenschaften:

$f_0$ = 10 Hz
A = 10
$\alpha$ = 1
C = 1 $\mu$F

*Drücken Sie* *Anzeige/Ausdruck*

10 [f] [a] 10 [A] 1 [f] [b] 1 [EEX]
[CHS] [B] [D] ⟶ 1.000–06 $C_1$

Wie aus diesem Beispiel zu ersehen ist, wird dem Benutzer die Möglichkeit an die Hand gegeben, auch ohne langwierige Vorarbeiten und ohne Programmierkenntnisse spezielle Aufgaben zu berechnen. Er kann so seine tägliche Arbeit weitaus zeitsparender, sicherer und genauer als bisher erledigen.

Nicht nur Lehrer werden an den Problemen Interesse haben, die der „Taschenrechner im Unterricht" mit sich bringt. Dieses Thema geht sozusagen alle an. Zwar ist anzunehmen, daß sich Pädagogen mehr als andere Leser für „Richtlinien" etc. interessieren. Der folgende Beitrag besteht aber zum überwiegenden Teil aus sehr nützlichen Diskussionen grundsätzlicher Rechnereigenschaften und gibt einen Katalog relevanter Auswahlkriterien, die praktisch bei jedem Kaufentscheid hilfreich sind. Spezielle Betrachtungen zum Einsatz im Mathematikunterricht und eine Reihe von Beispielen runden den Aufsatz ab.

A. Wynands

# Taschenrechner im Unterricht

„Noch nie habe ich bei Flächen- und Volumenberechnung mit einer 8. Klasse so viel geschafft wie im letzten Halbjahr. Der Taschenrechner ermöglichte ein schnelles Ermitteln vieler Ergebnisse. Formeln und Aufgabentypen wurden vermehrt eingeübt bzw. durchdrungen, Nebenrechnungen mit ‚glatten wie krummen' Zahlen verbrauchten kaum noch Zeit und Konzentrationsvermögen der Kinder."

Soweit sinngemäß die Äußerung eines Hauptschullehrers. Ein befreundeter Lehrer an einer gewerblichen Schule sagte mir kürzlich: „Vom Taschenrechner halte ich nichts. Die Schüler verlernen damit das (Kopf)-Rechnen. Zudem müßte ich ja bei Hausaufgaben und Klassenarbeiten viel mehr Aufgaben stellen und korrigieren, da die Schüler sonst zu schnell fertig sind."

Während man in nahezu allen Veröffentlichungen seit 1977 über den Taschenrechner in der Schule fast ausschließlich über das Wo, Wann und Wie schreibt, scheint mir die Diskussion über das Für und Wider in der *Schulpraxis* noch nicht beendet.

Dieser Beitrag soll neben einer kurzen Beschreibung einiger Teilbereiche des Ist-Zustandes Aussagen machen zu *Entwicklungstendenzen, Auswahlkriterien* und zu *spezifischen Einsatzbereichen* von Taschenrechnern im Unterricht. Dabei wird hier vorwiegend vom *einfachen Elektronischen Taschenrechner* gesprochen, wofür die Abkürzung ETR gewählt wurde. Auf programmierbare Taschenrechner (PTR) wird nur kurz eingegangen.

## 1 Zum Ist-Zustand des ETR im Unterricht

### 1.1 ETR als erlaubtes bzw. verbotenes Hilfsmittel in der Schule

Zur Zeit sind in allen Ländern der Bundesrepublik frühestens ab Klasse 7 ETR als *Rechenhilfsmittel* in Schulen durch entsprechende Erlasse zugelassen. 1975 wurde zunächst in Bayern der ETR im Unterricht, jedoch nicht bei Prüfungen genehmigt. 1978 ist wiederum Bayern das erste Land, in dem für das Schuljahr 1978/79 die Verwendung von Rechenstab und Logarithmentafel entfällt und der ETR die Rolle dieser Unterrichts-Hilfsmittel übernimmt.

Die entsprechenden Erlasse sind einerseits begrüssenswert, weil dadurch vielen Lehrern eine Rechts- und Orientierungshilfe gegeben wird. Andererseits scheinen folgende Punkte bedenkenswert:

- Zur Absicherung verbindlicher Richtlinien fehlen zur Zeit hinreichend aussagekräftige, empirische Untersuchungen. Es sollte die „Vorläufigkeit" dieser Erlasse erkannt werden.
- Durch diese vorläufigen Erlasse sollte sich kein „unvorbereiteter" Lehrer gezwungen fühlen, ETR ab Klasse 7 einzusetzen. Neuerungen in der Schule, die nicht vom Lehrer getragen werden, bedeuten eher Rück- als Fortschritte.
- Die Erlasse verbieten keineswegs explizit den kontrollierten Einsatz von ETR als *methodisches* Hilfsmittel vor Klasse 7. Ein persönlicher Freiraum für engagiertes Lehren muß auch bei aller Erlaßflut bleiben.

### 1.2 Richtlinien und Schulbücher

Orientierungshilfe und Richtschnur für den Unterricht sind in erster Linie die von der Kultusbürokratie und den Schulaufsichtsbehörden herausgegebenen Richtlinien und Lehrpläne und die hiernach ausgerichteten Schulbücher.

In Richtlinien und Lehrplänen nur weniger Bundesländer, z.B. Rheinland-Pfalz und Baden-Württemberg, wird z.Z. der Einsatz von ETR als Alternative zum Rechenstab ausdrücklich festgelegt.

In keinem vorliegenden Schulbuch[1] scheint mir eine befriedigende Einbeziehung des ETR weder als reines Rechenhilfsmittel noch als methodisches Hilfsmittel verwirklicht. Dies ist sicherlich im momentanen Schwebezustand nicht anders zu erwarten, der teils von Modernitätsdrang, teils durch starres Festhalten an alten Methoden und Hilfsmitteln (Rechenstab und Logarithmentafel) und teils durch Entwicklung und Abtesten von Unterrichtseinheiten und neuen Arbeitsmethoden und Beispielaufgaben geprägt ist. Man fragt sich jedoch, welchem Schüler damit gedient ist, wenn z.B. auf sieben Seiten der Umgang mit dem Rechenstab und danach auf *einer* Seite – fehlerbehaftet – der Einsatz des ETR „durchgenommen" wird.[2]

Ein Kapitel „Taschenrechner" sollte zugunsten eines durchgehenden Einsatzes von ETR an entsprechend zugeschnittenen Aufgaben vermieden werden.

### 1.3 Taschenrechner in der Primarstufe?

Nach einer im Herbst 1976 durchgeführten Befragung [14] von ca. 100 Lehrern der Klassen 5 bis 10 hielten nur 2% den Einsatz von ETR ab Klasse 5 für sinnvoll. Mehr als 80% befürworten den Einsatz ab Klasse 8.

Zum Einsatz von ETR als Rechenhilfsmittel im Primarbereich wird in den deutschsprachigen Veröffentlichungen – wenn überhaupt – kritisch-distanziert bis ablehnend Stellung genommen.

Ein Zwang zur Berücksichtigung von ETR in den Klassen 1 bis 4 aufgrund eines „de-facto-Benutzens" dieser Geräte durch die Schüler, scheint auch in Zukunft nicht gegeben zu sein. Einerseits wird nämlich ein sinnvolles Verbot in diesen Altersklassen weit eher befolgt als bei älteren Schülern, andererseits rechtfertigen hier keine Aufgabentypen den Einsatz eines elektronischen Rechenhilfsmittels.

Für den Fall, daß jedoch diese Prognose über den tatsächlichen Verwendungsgrad von ETR durch Primarstufen-Schüler falsch ist, werden von einer Projektgruppe TIM[1] in Münster Methoden und Arbeitsmaterialien erarbeitet zum methodischen Einsatz des ETR für 5- bis 12-jährige [7].

Dieser für Deutschland einzigartigen Aktivität stehen vielfältige, sicherlich kritischer zu verfolgende Projekte in den USA gegenüber. In [7] ist zu lesen: Regionale Schuldistrikte wie Philadelphia oder Fairfax Country/Virginia und lokale Schuldistrikte wie Broomfield/Colorado, Columbus/Ohio oder Western Springs/Illinois haben umfangreiche Unterrichtsmaterialien entwickelt, mit deren Hilfe der Taschenrechner *möglichst schnell* und *problemlos* in den Mathematikunterricht der Primarstufe (Klasse 1–6) integriert werden kann.

### 1.4 Verfügbarkeit von Schülern über ETR

Ein Zahlenbeispiel für die 9. Klasse ein und derselben Hauptschule, an der ich unterrichtete:

*Oktober 1974:* Von 18 Schülern einer freiwilligen Arbeitsgemeinschaft zum Thema Taschenrechner-Einsatz hatte 1 Schüler schon mal mit ETR gerechnet, *kein* Schüler hatte einen eigenen Rechner.

*Oktober 1976* (Prozentsätze der Schüler *aller* 9. Klasse):

96% hatten schon mit ETR gerechnet,
43% hatten einen eigenen ETR,
50% benutzen den ETR bei den Hausaufgaben als „Kontrollgerät".

*Februar 1979:*

100% hatten schon mit ETR gerechnet
95% hatten einen eigenen ETR
70% benutzten den ETR bei den Hausaufgaben „nicht nur als Kontrollgerät".

Ein Vergleich zweier Untersuchungen [14] und [5], bei denen ca. 6000 bzw. 5000 Schüler befragt wurden, zeigt, daß sich der Anteil aller Schüler der Klassen 7 bis 10, die über einen eigenen Taschenrechner verfügen, von 1976 bis 1978 von ca. 30% auf 60% in zwei Jahren verdoppelt hat. Der Anteil der Schüler,

---

1) Ansätze sind zu finden z.B. in den Unterrichtswerken „Gamma" für die Klassen 7 und 8 (Klett) und „Andelfinger/Nestle" (Herder).

2) vgl. Westermann-Mathematik (Neubearbeitung 1977) für Klasse 7.

1) TIM: Taschenrechner im Mathematikunterricht, Kontaktadresse: H. Meißner, Pädagogische Hochschule Münster, Fliednerstraße 21, D4400 Münster

die zu Hause über einen ETR verfügen, stieg in diesem Zeitraum von ca. 70% auf 85%. Es erscheint die Pauschalaussage richtig:

*1980 verfügt jeder Schüler ab Klasse 7 über einen ETR.*

## 2 Entwicklungstendenzen

### 2.1 Angebotsspektrum und Kosten

Einige Daten mögen die vorstehend angedeuteten Entwicklungstendenzen untermauern [6], [9]:

1970 erscheinen ETR auf dem Markt,
1972 kostet ein Bausatz für einen ETR mit [+], [−], [×], [÷], [=] etwa 500,– DM,
1975 werden ca. 7,5 Millionen ETR nach Deutschland eingeführt,
1972–1977 werden ca. 20 Millionen ETR in der Bundesrepublik verkauft,
1977 hat jeder zweite Bundesbürger zwischen 15 und 65 Jahren einen ETR.

Pikant mag die Information [6] sein, daß 1974 das Jahr war, in dem in der Bundesrepublik die meisten *Rechenstäbe* verkauft wurden, welche heute für die Schule bedeutungslos sind (vgl. 1.1 und Umfragen in [5], [14]).

Zur Preisentwicklung für ETR ist festzustellen, daß ein Umschwung von der Preiskonkurrenz zur Qualitätskonkurrenz eingesetzt hat. Der Preis eines ETR mit trigonometrischen und Exponential-Funktionen liegt heute bei dem eines (teureren) Schulbuches (z.B. TI 30 ca. 30,– DM). Es ist abzusehen, daß dies auch bald für entsprechende ETR mit **Flüssigkristall-Anzeige** (LCD) gilt, die für den Einsatz in der Schule wegen der problemlosen Energieversorgung wesentlich günstiger sind. Damit werden u.a. die bisher sinnvollen Ladekoffer für schuleigene Geräte überflüssig. Bei programmierbaren Taschenrechnern (PTR) sieht die Angebots- und Preisentwicklung ganz anders aus. Dazu einige Beispiele:

| Gerät | Zeitpunkt | Preis in DM | Bemerkungen |
|---|---|---|---|
| TI SR 52 | April 1976 | 1.150,– *) | |
| TI 57 | April 1979 | 90,– *) | empfohlene 128,– DM |
| TI 58 | April 1979 | 250,– | Hersteller- 348,– DM |
| TI 59 | April 1979 | 590,– | Preise: 798,– DM |
| HP 29 C | Februar 1978 | 575,– | |
| | März 1979 | 399,– | Angaben des Herstellers |
| HP 19 C | Februar 1978 | 948,– | |
| | März 1979 | 575,– | *) |
| | März 1979 | 498,– *) | Verkaufspreis im Handel (Aachen) |

Die Entwicklung auf dem Gebiet der Mikroprozessoren wird eine Annäherung der Leistungsfähigkeit von PTR an diejenige von Mikrocomputern bewirken. Um jedoch einer PTR-Euphorie, welche auf dessen Einsatz in einer „Schul-Informatik" abzielt, vorzubeugen, sei hier aus [4] zitiert:

„Die heute erhältlichen programmierbaren Taschenrechner, charakterisiert durch den Gebrauch von extrem niedrigen (auf **Assembler-Niveau** stehenden) Befehlen, sind eine vorübergehende Erscheinung."

„... Es ist sehr unwahrscheinlich, daß sich PTR in der heutigen Form halten werden. Bedeutend leistungsfähigere Systeme – möglicherweise auch solche, die einem Taschenrechner äußerlich sehr ähnlich sehen – werden an deren Stelle treten, Systeme, welche *höhere Programmiersprachen* und ein breites Spektrum von Dienstleistungen anbieten, (z.B.) **Editieren** und **Formatieren** von Texten und Programmen, **Simulationen**, rechner-unterstützter Unterricht, graphische Ausgabe, Programmierhilfen."

Zu untermauern sind solche Zukunftsperspektiven durch folgende Daten:

Von 1963 bis 1978 verringerten sich die Kosten pro **Schaltfunktion** auf ein Tausendstel. Die Anzahl der Transistorfunktionen auf 1 **LSI-Chip** mit einer Oberfläche von weniger als 1 $cm^2$ betragen ca.:

| 1972 | 1976 | 1977 | 1980 |
|---|---|---|---|
| 5000 | 15 000 | 30 000 | 100 000 – 1 000 000 |

**Flüssigkristall-Anzeige:** (LCD, *Liquid Crystal Display*). Immer häufiger verwendetes Anzeigeelement mit sehr geringem Stromverbrauch; sichtbar jedoch nur mit Fremdlicht.

**Assembler-Niveau:** Unterstes Programmiersprachen-Niveau, auf dem jeder einzelne Maschinenbefehl durch einen leicht merkbaren (*mnemonischen*) Ausdruck beschrieben wird.

**Editieren:** Schreiben eines Programms in einer Assembler- oder höheren Sprache.

**Formatieren:** Anordnung von Datenzeichen, Programmbefehlen etc. nach einem fest vereinbarten Schema.

**Simulation:** Darstellung eines realen Systems mit einem *Modell*, das im äußeren Verhalten und den internen Abläufen dem wirklichen System entspricht und das bequeme Studium dieses Systems erlaubt.

**Schaltfunktion:** (Z.B. Transistorstufe (Inverter), UND, ODER, NAND (auch: *Gatterfunktion*).

**LSI-Chip:** Halbleiterplättchen mit mehr als 100 Schaltfunktionen (LSI: *Large Scale Integration*).

### 2.2 Die GDM-Stellungnahme

Eine grundsätzliche Stellungnahme professioneller Fachdidaktiker in der Gesellschaft für Didaktik der Mathematik (GDM) zum Einsatz von ETR im Mathematikunterricht ist in [10] enthalten. Hierhin heißt es u.a.:

1. Als Rechenhilfsmittel sollten ETR – anstatt Rechenstab oder Logarithmentafel – ab dem 7. Schuljahr generell zugelassen werden.
2. Keinesfalls wird den Schülern in den ersten Schuljahren die Aneignung der grundlegenden Rechenoperation durch ETR abgenommen. Der Gefahr einer Abhängigkeit von ETR ist durch verstärktes Kopf- und Überschlagsrechnen sowie durch Aufzeigen der Grenzen des ETR entgegenzuwirken. Der ETR ermöglicht experimentelle Schüleraktivitäten und liefert für zahlreiche Begriffsbildungen eine konkrete numerische Ausgangsbasis (vgl. hierzu Abschnitt 4.2).
3. Im Rahmen der Lehreraus- und -fortbildung sind Grundkenntnisse über die Funktionsweise der Rechner und methodische Möglichkeiten zu vermitteln, wodurch Schüler zu einem verständigen und kritischen Umgang mit Taschenrechnern geführt werden können.
4. In langfristigen Forschungs- und Entwicklungsarbeiten sind Formen und Konsequenzen des Einsatzes von ETR für mathematische Lernprozesse auf allen Stufen zu untersuchen. Hierfür sind Arbeitsmaterialien zu entwickeln und zu erproben. Erst nach Vorliegen gesicherter Erfahrungen sollte eine etwaige Änderung von Stoffplänen und Richtlinien erfolgen.

Neben der Betonung des Stellenwertes von ETR, der in Abschnitt 4 punktuell konkretisiert werden soll, werden deutlich abnehmende (Kopf-)Rechenfertigkeit und Rechnergläubigkeit als Gefahren des ETR-Einsatzes gesehen. Vorliegende Untersuchungen, wie z.B. [1], [11], sind durch langfristige mit größeren Probandenzahlen zu ergänzen.

Diesen Gefahren sei hier eine weitere hinzugezählt: Die sicherlich durch die sich neu erschließenden Möglichkeiten beim Einsatz von ETR, PTR und Computern aufgekommene richtige These, wonach die gesamte (Schul-)Mathematik vom algorithmischen Standpunkt aus neu zu durchdenken ist [2], [3], darf nicht fehlinterpretiert werden und das Kind (strukturorientierte Mathematik) mit dem (Algorithmen-)Bade ausschütten.

## 3 Auswahlkriterien und Test für ETR im Unterricht

Nach den allgemeinen Betrachtungen sollen in diesem Abschnitt einige konkrete Informationen zur Auswahl von Taschenrechnern und zum Testen ihrer Eigenschaften gegeben werden. Diese sollen sowohl vor dem Kauf und beim Überprüfen eines neuen ETR hilfreich sein, als auch beim Kennenlernen der Funktionseigenschaften eines „fremden" ETR.[1]

Hierdurch soll zudem das häufig sehr mühsame Durcharbeiten der einzelnen Bedienungsanleitung für den ersten ETR-Einsatz erleichtert und z.T. sogar ersetzt werden.

### 3.1 Tastatur

- Alle Tasten sollen *hinreichend groß* und bedienungsfreundlich sein.
- *Farbige Tastenblöcke* sind optische Bedienungshilfen.
- *Sensortasten* mit Piepston ([♪]-Taste) können im Unterricht störend sein; ein spürbarer *Druckwiderstand* beim Betätigen einer Taste ist eine gute Kontrollmeldung für den Benutzer.
- *Prellfrei* müssen alle Ziffern- und Funktionstasten sein.
  Test: Bei *einmaligem* kurzem, langem, schrägem oder auch zittrigem Tippen einer Ziffern- (Funktions-)Taste darf die jeweilige Ziffer (Funktion) nur *einmal* aufgerufen werden.
- *Mehrfachbelegungen* von Tasten sind bei leistungsfähigeren ETR häufig nicht zu vermeiden; der „Störeffekt" ist auch in niedrigen Klassen (5–8) dann gering, wenn alle Tasten einer Grundausstattung (vgl. 3.8) direkt, d.h. ohne vorherigen Aufruf einer zusätzlichen „Umschalttaste" wie [F] oder [2nd] anwählbar sind.

### 3.2 Anzeige

Einige Auswahlkriterien für ETR stehen in Zusammenhang mit der Farbe der angezeigten Ziffern [12]:

[1] *Anmerkung:* Die folgende Liste wird längst nicht auf alle ETR-Typen eingehen können; einige Kriterien sind zudem sicherlich „subjektiv gefiltert". Der Leser möge dies bedenken und gegebenenfalls entschuldigen. Eine ausführlichere Darstellung findet man in [8] und [12].

| Anzeige Farbe | Anzeige Art | Energie-verbrauch | Empfind-lichkeit (Stoß, Fallen) | Ablesemöglichkeit Leuchtstärke | Ablesemöglichkeit Ziffern-größe | Beleuchtungs-abhängigkeit |
|---|---|---|---|---|---|---|
| rot | Leucht-dioden | groß | gering | gut } selbst-leuch-tend | klein | je heller die Umgebung, umso schlechter ablesbar |
| grün oder blau | Digitron Anzeige | gering | groß | gut } selbst-leuch-tend | groß | |
| grau (farb-los) | Flüssig-Kristalle | sehr klein | mittel | *nicht selbst-leuchtend* | groß | nur im Dunkeln nicht lesbar |

In **Fließkommadarstellung** sollte der ETR über eine mindestens 8-stellige Anzeige mit *Vorzeichen-Stelle* verfügen. *Test:* Kontrollieren Sie im Kopf

a) $2 \cdot 9\,999\,999 = 19\,999\,998$

b) $0{,}2 \cdot (-0{,}999\,999\,9) = -0{,}199\,999\,98$

und jetzt mit ETR z.B.

$\underbrace{\boxed{\cdot}\,2}_{0{,}2}\ \boxed{\times}\ \underbrace{\boxed{\cdot}\ 999\,999\,9\ \boxed{+/-}}_{(-0{,}999\,999\,9)}\ \boxed{=}$ ?

Eine *rechtsbündige Anzeige,* bei der die Ziffer mit dem niedrigsten (Dezimal-) Stellenwert am rechten Rand der Anzeige erscheint, ist bis zur 7. Klasse einer linksbündigen Anzeige vorzuziehen, da bei Additon und Subtraktion ganzer Zahlen die Ziffern stellenwertgerecht übereinander im x- und y-Register registriert scheinen.

*Beispiel:* Nach 1234 $\boxed{+}$ 567 ist

im y-Register $\boxed{1234}$

und im x-Register $\boxed{567}$

Leistungsstärkere ETR haben eine **Scientific Notation** oder *exponentielle Zahldarstellung.*

Diese erkennt man an der Taste $\boxed{EE}$ (*Enter Exponent*).

*Test:* 2 000 000 $\boxed{\times}$ 3 000 000 $\boxed{=}$ ?

ETR mit $\boxed{EE}$ -Taste zeigen: $\boxed{6.\ 12}$ (= $6 \cdot 10^{12}$ gleich „6 mit 12 Nullen").

ETR ohne $\boxed{EE}$ -Taste sollten das Überschreiten der Anzeigekapazität (*Overflow*) anzeigen und nach Möglichkeit die ersten 8 wesentlichen Ziffern richtig anzeigen, z.B. $\boxed{\sqcap 60\,000.\ 000}$.

Sehr störend – besonders für Schüler bis zur 8. Klasse – sind ETR mit der Taste $\boxed{EE}$, welche z.B. für 4 : 5 als Ergebnis $\boxed{8.\text{-}01}$ anzeigen: vgl.

$4 : 5 = 0{,}8 = 8 \cdot 10^{-1} \mathrel{\hat{=}} 8.\text{-}01.$

### 3.3 Notation

Hierbei sind generell ETR mit *algebraischer Notation* (AN) und solche mit *umgekehrter polnischer Notation* (UPN)[1] zu unterscheiden.

Erkennungsmerkmal: UPN-Rechner haben *keine* „Ergebnistaste" $\boxed{=}$. Sie besitzen jedoch eine besondere Eingabe- (oder *Enter*-)Taste $\boxed{ENTER}$ oder $\boxed{\uparrow}$.

Die schuladäquate Notation dürfte in der Sekundarstufe I wohl die algebraische sein, bei der einfache Aufgaben zu den Grundrechenarten so eingetippt werden, wie man diese schreibt und liest. Manche ETR mit algebraischer Notation verfügen, um rechentechnische Nachteile gegenüber UPN-Rechnern auszugleichen, über eine „Hierarchie"-Eigenschaft (ANH), wodurch z.B. die Regel „Punkt-vor-Strich-Rechnung" beachtet wird. Bei solchen ETR darf man sinnvollerweise nicht auf Klammer-Tasten $\boxed{(}$ $\boxed{)}$ verzichten.

*Tests:* Kontrollieren Sie im Kopf:

$2 + 3 \cdot 4 = 14;\ (2 + 3) \cdot 4 = 20;\ 2 + 3 \cdot \sqrt{4} = 8$

**Fließkommadarstellung:** Bei der Zahleingabe wird das Komma genau dann gesetzt, wenn die Komma- oder Dezimalpunkt-Taste $\boxed{\cdot}$ gedrückt wird. Bei der Zahlverarbeitung *fließt* das Komma automatisch an die richtige Stelle.

**Scientific Notation:** (Wissenschaftliche Notierung). Zur Darstellung sehr kleiner und großer Zahlen geeignet, wobei das Komma sich immer automatisch hinter die erste Stelle setzt. Gilt für eine Zahl $x = m \cdot 10^h$, so zeigt der ETR hierfür m und rechts daneben h an.

[1] Die Notation oder Term-Schreibweise geht auf den polnischen Mathematiker *Jan Lukasiewicz* (1878–1956) zurück, wonach man jeden (aussagenlogischen) Term ohne „Klammern" notieren kann.

und mit ETR

2 [+] 3 [x] 4 [=] es erscheint < 20 bei AN / 14 bei ANH

2 [+] 3 [=] 5 [x] 4 [=] < 20 bzw.

[(] 2 [+] 3 [)] 5 [x] 4 [=] < 20

2 [+] 3 [x] 4 [√x] [=] < 10 bei AN / 8 bei ANH / 4,47 ... = $\sqrt{(2+3)\cdot 4}$

Von Rechnern, die diese Aufgabe wie $\sqrt{(2+3)\cdot 4}$ rechnen, ist unbedingt abzuraten.

Bei ETR mit UPN z.B.:

2 [ENTER] 3 [ENTER] 4 [x] 12 [+] < 14

2 [ENTER] 3 [+] 5 4 [x] < 20

2 [ENTER] 3 [ENTER] 4 [√x] 2 [x] 6 [+] < 8

### 3.4 Konstantenautomatik

Viele ETR verfügen über eine häufig sehr vorteilhafte Konstantenautomatik. Bei Rechnern, die hierfür *keine* besondere Taste z.B. [K] haben, sollte einheitlich jeweils die *zweite* Zahleingabe zusammen mit dem Operationssymbol als „Operator" gespeichert werden.

*Tests* für ETR ohne spezielle Konstantentaste:

Rechnen Sie zuerst gemäß: 2 [x] 3 [=] < 6

und danach: 4 [=] < ?

Erscheint jetzt 4, so hat der ETR *keine* Konstantenautomatik;

12, dann wurde (x3) gespeichert („mal-3-Maschine");

8, dann wurde (2x) gespeichert („2-mal-Maschine").

Testen Sie auch mit [+] [−] [÷] und ggf. [$y^x$] statt [x].

*Tests* für Rechner mit Konstantentaste [K]:

zuerst: 2 [*] [K]

dann 6 [=] <
- 8 = 6 + 2, für [*] = [+]
- 4 = 6 − 2, für [*] = [−]
- 12 = 6 · 2, für [*] = [x]
- 3 = 6 : 2, für [*] = [÷]

Rechnen Sie nun noch gemäß folgender Anweisung. Beobachten Sie nach jedem Tastendruck, was in der Anzeige erscheint und entdecken Sie so, ob der ETR über eine spezielle Konstantenautomatik verfügt:

2 [*] ? [=] ? [=] ? [=] ? ... wählen Sie

[*] ∈ {[+], [−], [x], [÷], [$y^x$]}

### 3.5 Speicher

Jeder ETR sollte mindestens über einen „**Schreib-Lese-Speicher**" zum Aufbewahren von Zwischenergebnissen oder Konstanten verfügen. Rechner mit nur einem Speicher erkennt man z.B. an den Tasten [M] („Merker", *Memory*) oder [x → M]. Der Rückruf (*Recall*) des Speicherinhalts erfolgt mit [MR], [RM] oder [M → x].

*Test:* Kontrollieren Sie im Kopf:

$$\frac{2\cdot 3+8}{4\cdot 5-13}=\frac{2\cdot 3+8}{20-13}=\frac{2\cdot 3+8}{7}=\frac{6+8}{7}=\frac{14}{7}=2$$

und mit ETR:

4 [x] 5 [−] 20 13 [=] 7 [M] 2 [x] 3 [+] 6 8

[=] 14 [÷] [MR] 7 [=] < 2

Rechner mit ausschließlich „saldierendem" Speicher, bei denen auf Tastendruck [M+] (bzw. [M−]) automatisch zum (vom) Speicherinhalt der angezeigte Wert addiert (subtrahiert) wird, sollten über eine separate Speicherlöschtaste [CM] (*Clear Memory*) verfügen. Zur Vermeidung von fehlerhaftem Speichereinsatz ist es vorteilhaft, wenn das „Belegtsein" des Speichers im Anzeigefeld durch ein Symbol (z.B. (M)) angezeigt wird. Zur Kontrolle, was abgespeichert wurde, ist auch eine Speicheraustauschtaste (z.B. [x ↔ M] oder [E X C]) vorteilhaft. Die Tastenfolge [x ↔ M] [x ↔ M] bewirkt keine Zustandsänderung des Rechners, eine angefangene Rechnung kann danach „ungestört" fortgesetzt werden.

ETR mit mehreren Speichern erkennt man i.d.R. an den Tasten [STO] (*Store*) und [RCL]. Die Speicher sind durchnumeriert: z.B. 00, 01, 02, ... 29 oder 0, 1, ... 7.

*Test:* zuerst 10 [STO n];

d.h. Eingabe von 10, Drücken von STO und Eingabe der Ziffer(nfolge) von n. Bei Fehlermeldung, z.B.

blinkende Anzeige, ist Speicherplatz n nicht mehr vorhanden.

Dann 2 [x] [RCL n] [=] ⟨ 20.

Sinnvoll erscheint ein **„Nur-Lese-Speicher"** (*Read Only Memory:* ROM) für einen Näherungswert von $\pi$ ($\approx$ 3,141 592 654) [$\pi$].

*Test:*

3 [$x^2$] [x] [$\pi$] [=] ⟨ 28,27 ... (= $3^2 \cdot \pi$).
3,14 ...

### 3.6 Hinweise zu weiteren Funktionstasten

[%] Die Prozentautomatik ist für (allgemeinbildende) Schulen *nicht* zu empfehlen. Sie arbeitet bei unterschiedlichen ETR-Typen sehr unterschiedlich, kann zum unverständigen „Rezeptgebrauch" und in eine didaktische Sackgasse führen [9], [13]. Hilfreich als „Prothese" ist die Prozentautomatik jedoch z.B. bei äußerst rechenschwachen Schülern. Hierzu wird in [1] über Ergebnisse in Jungarbeiterklassen berichtet.

[$y^x$] *Tests:* 2 [$y^x$] 3 [=] ⟨ 8 (= $2^3$);

Merke: *Zuerst die Basis eingeben.*

2 [$y^x$] 3 [+/-] [=] ⟨ 0,125

$\left(= 2^{-3} = \frac{1}{2^3} = \frac{1}{8}\right)$

8 [$y^x$] 3 [1/x] [=] ⟨ 2

(= $8^{(1/3)} = \sqrt[3]{8}$)

[$\pi$] [$y^x$] [$\pi$] [=] ⟨ 36,4 ... (= $\pi^\pi$);

Manche ETR liefern hier falsche Werte!

2 [+/-] [$y^x$] 5 [=] ⟨ Fehleranzeige o. „blinkende" 32

($(-2)^5 = -32$)

*Merke:* Die Basis (oder „Grundzahl") muß bei Verwendung von [$y^x$] positiv sein, da der ETR zur Berechnung des Funktionswertes den Logarithmus der Basis verwendet.

[$\sqrt[x]{y}$] *Test:* 8 [$\sqrt[x]{y}$] 3 [=] ⟨ 2 (= $\sqrt[3]{8}$)

Merke: *Zuerst den Radikanden eingeben.*

[$10^x$] *Test:* 3 [$10^x$] ⟨ 1000 (= $10^3$)

3 [+/-] [$10^x$] ⟨ 0,001 $\left(= 10^{-3} = \frac{1}{10^3}\right)$

3 [$\sqrt{x}$] [$10^x$] ⟨ 53,9 ...

(= $10^{\sqrt{3}} = 10^{1,73\ldots}$)

[$e^x$] *Test:* 1 [$e^x$] ⟨ 2,718 281 82 ...

(= $e^1$ = e; Eulersche Zahl)

2 [$e^x$] [$e^x$] ⟨ 1618,1 ... (= $e^{(e^2)}$)

[log] *Test:* 100 [log] ⟨ 2 (= log 100)

Dekadischer oder Zehner-Logarithmus

123 [log] [$10^x$] ⟨ 123 (= $10^{\log 123}$)

1,23 [+/-] [$10^x$] [log] ⟨ − 1,23

(= $\log(10^{-1,23})$)

[ln] *Test:* 10 [$e^x$] [ln] ⟨ 10 (= $\ln(e^{10})$)

Bei ETR mit trigonometrischen oder Winkel-Funktionen sollten Winkel sowohl im (Alt-) Grad- als auch im Bogenmaß einzugeben sein. Der jeweilige *Winkelmodus* sollte direkt der Stellung eines entsprechenden Schiebeschalters oder besser noch einem einprägsamen Symbol, z.B. GRAD bzw. RAD, im Anzeigefeld zu entnehmen sein. Zudem sollten auch Winkelfunktionswerte für Winkel grösser als 360° bzw. 2 $\pi$ angezeigt werden.

*Tests:*

90 [sin] ⟨
- 1 wenn *Altgrad*-Einstellung
- 0,98 ..., wenn *Neugrad*-E. (90° ≙ 100g)
- 0,89 ..., wenn *Bogenmaß*-E. $\left(90° \triangleq \frac{\pi}{2}\right)$

3660 [cos] ⟨
- 0,5 ... (= cos 3660° = cos 60°)
- 0,587 ... (= cos 3660g = cos 60g)
- −0,9990 .. (= cos 3660 ≈ cos 3,186)

Zur Berechnung von Cotangens- und Arcus-Cotangens-Werten beachte man:

$$\cot x = \frac{1}{\tan x} \quad \text{und} \quad \operatorname{arccot} x = \arctan\left(\frac{1}{x}\right)$$

**Schreib-Lese-Speicher:** (RAM, *Random Access Memory*). Speicher, in den beliebig neu eingeschrieben werden kann. Beim Lesen werden die Speicherinhalte nicht verändert, beim Abschalten der *Energie*versorgung aber in der Regel gelöscht.

**Nur-Lese-Speicher:** (ROM, *Read Only Memory*). Speicher, der während der „Programmierungsphase" beschrieben wird, danach nur noch ausgelesen werden kann und beim Abschalten der Energieversorgung nicht gelöscht wird.

*Tests:* (Winkelmodus *Altgrad*)

10 [tan] [1/x] 5,67 ... (= cot 10°)
[cot]

5,67 [1/x] [arc] [tan] 10,0 ... (= arc cot 5,67)
[arc cot]

### 3.7 Rechengenauigkeit

Rechner ohne Rundungsautomatik liefern wegen der prinzipiell ungenauen oder fehlerbehafteten digitalen Darstellung der „meisten" reellen Zahlen häufig „offensichtlich falsche" Ergebnisse, z.B.

a) 3 [1/x] [1/x] 3,000 000 3 $\left(\text{vgl. } \frac{1}{\frac{1}{3}} = 3\right)$

b) 2 [÷] 789 [x] 789 [=] 1,999 957 2
$\left(\text{vgl. } \frac{2}{789} \cdot 789 = 2\right)$

c) 2 [√x] [x²] 1,999 999 8 (vgl. $(\sqrt{2})^2 = 2$)

Solche Fehler vermeiden ETR mit „5-Strich-4-Rundungsautomatik", wobei die letzte angezeigte Ziffer „aufgerundet" wird, wenn die erste nicht mehr angezeigte Ziffer einen Wert größer oder gleich 5 hat.

*Test:* 5 [÷] 9 [=] 0,555 ... 5 ⇒ keine Rundung,
0,555 ... 6 ⇒ mit $^5/_4$-Rundung.

Durch das Abschneiden (*Underflow*) von Ziffern mit geringerem Stellenwert treten vorwiegend bei ETR ohne exponentielle Zahldarstellung ( [EE] -Taste) häufig „Null-Katastrophen" auf. *Beispiel:*

a) 1 [÷] 123 [÷] 123 [÷] [x] 123
0,00813 0,000066
[x] [x] 123 [=] 0,998514
0,008118

b) 1 [÷] 789 [÷] 789 [÷] 789 [x] 789 [x]
0
789 [x] 789 [=] 0 (statt 1)

ETR mit exponentieller Zahldarstellung vermeiden solche „katastrophalen" Fehler durch automatisches Umformen der Zahldarstellung: Das Komma „gleitet" so, daß jeweils möglichst viele „wesentliche" Ziffern erfaßt werden.

Die prinzipielle Beschränktheit auch solcher ETR zeigt das *Beispiel:*

0,2 [x²] [x²] [x²] [x²]
0,04 0,0016 0,00000256 $6{,}5 \ldots \cdot 10^{-12}$
[x²] [x²] [x²]
$4{,}2 \ldots \cdot 10^{-23}$ $1{,}8 \ldots \cdot 10^{-45}$ $3{,}4 \ldots \cdot 10^{-90}$
[x²]
$1 \cdot 10^{-99}$ und Fehlermeldung

### 3.8 Empfehlungen

Unter Beachtung vorstehender Kriterien und Hinweise sei hier für den Einsatz in der Schule empfohlen:

*Wenn eine „Grundausstattung"* bis zur 9. Klasse anzuschaffen ist, dann: ETR mit algebraischer Notation (eventuell mit Klammern und Hierarchie) und folgenden Tasten:

[+] [−] [x] [÷] [1/x] [√x] [+/−] [π] [M] [MR] [C] [CE]

(Löschen der letzten Eingabe),

Konstantenautomatik ( [K] ).

Rundungsautomatik und Speicheraustauschtasten [x ↔ y] [x ↔ M] sind vorteilhaft.

Aus energietechnischen und organisatorischen Gründen sind ETR mit Flüssigkristallanzeige (LCD) und Abschaltautomatik besonders geeignet.

Wenn *nicht nur* Grundausstattung, dann *zusätzlich*:

[yˣ] [eˣ] [ln] ... [sin] ... [arc] ... und [EE].

Für einen Frontalunterricht eignen sich sehr gut Taschenrechner-Demonstratoren, wie sie bisher von der Firma ARISTO vertrieben wurden. Es ist zu hoffen, daß solche Geräte – Maße 74 cm × 37 cm × 4,5 cm; 2,1 cm hohe Anzeige; Preis: ca. 300,– DM – die in ihren Funktionen vollständig den Taschen-Rechnern entsprechen, bald verstärkt auf dem Markt sein werden.

## 4 Fachdidaktische Komponenten des ETR im Mathematikunterricht

Die Vorzüge des ETR als schneller, zuverlässiger Rechner im naturwissenschaftlichen und/oder berufsbezogenen Unterricht scheinen mir offensichtlich. Hier sollen Beispiele in vier Anwendungsgebieten skizziert werden, in denen der ETR als fachdidaktisch begründetes methodisches Hilfsmittel einzusetzen ist. Diese Beispiele beziehen sich auf den Mathematikunterricht der Sekundarstufen allgemeinbildender Schulen, vgl. [15].

### 4.1 Sachrechnen

Sachrechnen und Lösen numerischer Probleme sind prädestinierte Einsatzbereiche des ETR im Unterricht, wenn man bedenkt, daß hierbei nicht Rechenfertigkeit sondern Rechenfähigkeit im Sinne von Analysieren und Mathematisieren umweltbezogener Situationen und das Interpretieren numerischer Ergebnisse zu üben sind.

Der ETR ist hierbei ein Lehr- und Lernmedium

a) zur „Verbreiterung der Übungsphase" in dem Sinne, daß wesentlich mehr Beispiele in gleicher Zeit bearbeitet werden können und
b) für eine (maßvolle!) Stofferweiterung, die vorwiegend solche Gebiete betreffen sollte, welche umweltrelevant sind, aber wegen des Rechenaufwandes bisher gemieden wurden (z.B. Wachstumsprozesse, beschreibende Statistik, (lin.) Optimierungsprobleme).

Folgende Punkte erscheinen hier angebracht:

- Eine Vielzahl konkreter, umweltrelevanter Aufgaben zur gleichen Problemklasse können mit ETR durchgerechnet werden, wodurch sich die Kenntnis der entsprechenden Formel bzw. des Rechenschemas festigt.
- Mehr Problemfelder und zugehörige Lösungsmethoden können behandelt werden, wodurch das Repertoire von Lösungsstrategien vergrößert wird.
- Das selbsttätige Erstellen von Tabellen und Graphiken wird möglich, wodurch diese für den Schüler durchschaubarer werden und „funktionales Denken" geübt wird.
- Rechenschwache intelligente Schüler werden grössere Erfolgserlebnisse haben, weil dem „Rechnen" im Mathematikunterricht der ihm zustehende Stellenwert deutlich zugewiesen wird.
- Rechenschwache und weniger intelligente Schüler verbrauchen ihre Konzentrationsfähigkeit nicht durch Rechenballast.
- Alle Schüler erhalten mehr Chancen, umweltrelevante Mathematik kennenzulernen.

*Anmerkung:* Mit Sicherheit vergrößern wird sich durch Einbezug von ETR der „Leistungsabstand" zwischen den (rechenstarken) intelligenten Schülern und den „Rechenwundern" ohne Mathematikverständnis.

Beispiele und Methoden zum Taschenrechnereinsatz im Sachrechnen findet man auf mehr als 60 Seiten in [12].

### 4.2 Numerische Ausgangsbasis für abstrakte Begriffe

Zur Erarbeitung von Begriffen mit dem ETR dienen u.a.

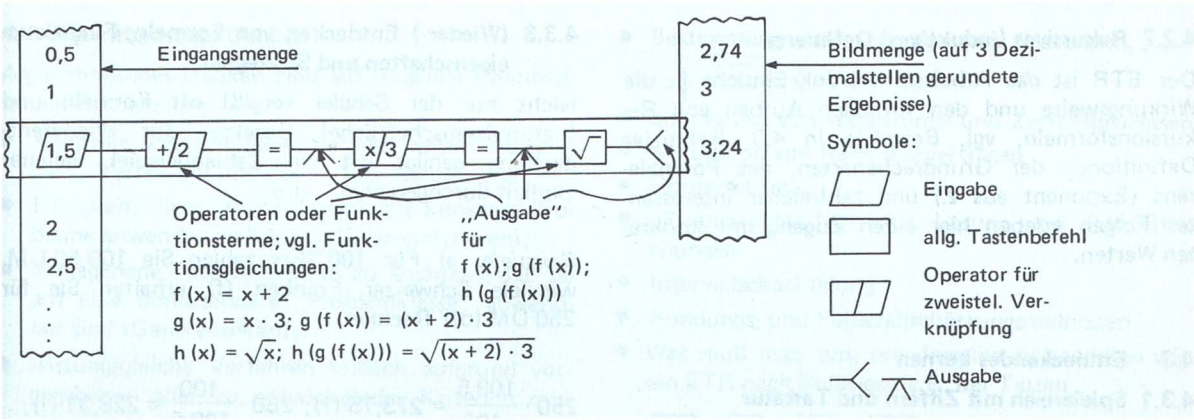
- das Beobachten der Zahlen im Anzeigefeld,
- das Planen, Erstellen und Ausfüllen von Wertetabellen und graphischen Darstellungen,
- die manuelle Tastenbetätigung besonders bei sich „wiederholenden" Rechenablaufteilen,
- ein bildhaft-symbolisches Protokoll dieses manuellen Handlungsablaufs in Form von Rechenablaufplänen, vgl. [13].

Zur Vertiefung und Festigung von Begriffen trägt der ETR-Einsatz durch weniger mühevolles und schnelles Bearbeiten vieler Beispielaufgaben bei, wodurch „Begriffsumfang" und „Begriffsinhalt" besser ausgeschöpft und abgegrenzt werden können.

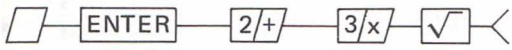
*Beispiele:*

#### 4.2.1 Operator und Funktionen

Operatoren als Abbildungsvorschriften erfahren in Schulbüchern häufig eine ikonische Veranschaulichung durch fiktive, nicht reale Maschinen. Dagegen ist der ETR eine wirklich funktionierende Operatormaschine (*Black Box*) für Zahlen. Zur Hinführung und Darstellung der Begriffe Operator, Funktion, Hintereinanderausführung (Verkettung) von Funktionen zeichne man einen Rechenablaufplan (RAP, in dieser oder ähnlicher Form) auf eine Transparentfolie, die man so über die „Elemente der Eingangsmenge" schiebt, daß diese nacheinander die „Leerstelle" im RAP belegen:




Man beachte, daß die RAP vom jeweilig verwendeten ETR-Typ abhängig sind. Für ETR mit UPN gilt z.B.:




vgl. $f(x) = 2 + x$; $g(f(x)) = 3 \cdot (2 + x)$;

$h(g(f(x))) = \sqrt{3 \cdot (2 + x)}$

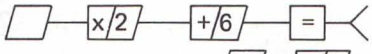
### 4.2.2 Eigenschaften von Funktionen

Mit dem Rechenhilfsmittel ETR können schnell Wertetabellen und graphische Darstellungen für Funktionen erstellt werden. Damit schafft der ETR die Voraussetzung zum „Beobachten" von Zahlenfolgen und Kurvenverläufen, welche eine konkrete numerische bzw. graphische Ausgangsbasis für folgende Begriffe ist:

Monotonie („je mehr desto mehr" bzw. „... desto weniger"); steigender, fallender, gleichbleibender Kurvenverlauf; Grenzwert; beschränkt, konvex, konkav, stetig, differenzierbar.

### 4.2.3 (Un)-Gleichungen als Aussageformen

*Beispiel:* Peter rechnet so:




Monika rechnet so:

Welche der Zahlen 1, 2, 3, 4 bzw. 5 müssen beide gleichzeitig eingeben (als Belegung der Leerstelle ▱ ), wenn Peters Ergebnis größer (gleich bzw. kleiner) sein soll als Monikas?

### 4.2.4 Zahlbegriffsbildung, Intervallschachtelung und Umkehrfunktion

Der Begriff „reelle Zahl" als Zentrum einer Intervallschachtelung ist mit ETR für jeden Schüler leicht erfahrbar, weil „die Schüler selbst tun können, wovon früher der Lehrer nur erzählte" (wie *A. Kirsch* es ausdrückt). Durch eigenaktives Aufsuchen von Lösungen, z.B. für $x^2 = 3$; $x^3 = 2$; $1{,}05^x = 2$; $\sin x = 0{,}5$, können nicht nur Schachtelungsstrategien (u.a. gezielte Probiermethoden) sondern auch der Begriff der Umkehrfunktion (bzw. -relation) empirisch vorbereitet und gefestigt werden.

### 4.2.5 Zahldarstellung

Die in der Regel dezimale Zahldarstellung im ETR-Anzeigeregister führt augenfällig unmittelbar zur Dezimalzahldarstellung (DZD) von Zahlen (Bruchzahlen) und mittelbar zu den Begriffen (nicht-) abbrechende, periodische DZD, Periodenlänge und Länge der Vorperiode. Das bei Schülern häufig beliebte Spielen mit „sehr großen" und „ganz kleinen" Zahlen regt an zur Suche nach Darstellung und Verarbeitungsmöglichkeit solcher Zahlen (Abspalten von 10er Potenzen, Exponentialdarstellung [EE]-Taste).

### 4.2.6 Fehler und Fehlerfortpflanzung

Der kritische Umgang mit ETR führt unmittelbar zu den Begriffen *subjektive Fehler* (z.B. Bedienungsfehler des Menschen) und *objektive Fehler* („Gerätefehler"). Letztere führen zur Erfahrung der Genauigkeit und Grenzen des ETR und zu den Begriffen absoluter und relativer Fehler einer digitalen Zahldarstellung oder einer gemessenen Größe. Beim Rechnen mit gemessenen Größen ist für den Schüler die Auswirkung von Fehlern, d.h. die *Fehlerfortpflanzung* offensichtlich, wenn er mit den kleinst- bzw. größtmöglichen Werten eine „Doppelrechung" durchführt und damit die sicheren Stellen des Ergebnisses ermittelt.

### 4.2.7 Rekursives (induktives) Definieren

Der ETR ist *das* Arbeitsmittel zur Einsicht in die Wirkungsweise und den formalen Aufbau von Rekursionsformeln, vgl. Beispiele in 4.3. Rekursive Definitionen der Grundrechenarten, des Potenzierens (Exponent aus $\mathbb{Z}$) und zahlreicher interessanter Folgen erleben hier einen Zugang mit konkreten Werten.

## 4.3 Entdeckendes Lernen

### 4.3.1 Spielereien mit Ziffern und Tastatur

Berechne, beobachte, vermute, begründe oder verwerfe die Vermutung:

a) $1 \cdot 1 =; 11 \cdot 11 =; 111 \cdot 111 =; \ldots$

b) $1{,}1^3 =; 1{,}01^3 =; 1{,}001^3 =; \ldots$

c) Berechne die Differenz von n-stelligen Zahlen und deren Spiegelzahl (z.B. $581 - 185 = 396 = 4 \cdot 99$).

d) Wähle eine Zahl. Bilde die Summe dieser Zahl und deren Spiegelzahl. Mit dem Ergebnis mache das gleiche bis ein Palindrom (Zahl gleich Spiegelzahl) entsteht.
Z.B.: $159 \rightarrow (159 + 951) \rightarrow 1110 \rightarrow (1110 + 0111) \rightarrow 1221$.
Wer beweisen kann, daß dieses Verfahren immer ein Palindrom liefert, sollte die (bisher nicht bewiesene) Vermutung als Satz veröffentlichen.

e) Berechne $2^n$, $3^n$, $4^n$, $5^n$ soweit der ETR genaue Ergebnisse liefert. Beobachte die letzte Ergebnisziffer.

f) „Drehzahl-Wörter": Gib in den ETR die Zahl 35 137,135 ein, drehe den Rechner um 180° und lies (SEI·LEISE). Ergänze: $9 \cdot 19 \cdot 43$ [+] [ ] [=] so, daß entsprechend LIES·ESEL erscheint. Weitere Beispiele in [12].

### 4.3.2 Geometrie

*Beispiel:* Alle Schüler zeichnen mehrere Rechtecke und messen die Seiten (a und b) und die Diagonale (d). Die (gerundeten) Maßzahlen a, b und d werden tabelliert und es werden „charakteristische Beziehungen" (allgemeingültige Aussagen, Invarianten) gesucht, z.B. $a + b > d$, $a^2 + b^2 \approx d^2$. Analoges Vorgehen ist z.B. bei den Strahlensätzen, bei der Definition von Trigonometrischen Funktionen, bei der Suche nach $\pi$ aber auch bei sehr vielen naturwissenschaftlichen Gesetzen (Versuchsauswertung) angebracht.

### 4.3.3 (Wieder-) Entdecken von Formeln, Funktionseigenschaften und Strategien

Nicht nur der Schüler vergißt oft Formeln und (naturwissenschaftliche) Gesetze. Zur Wiederentdeckung genügt oft ein Zahlenbeispiel, welches explizit durchgerechnet wird.

*Beispiele:* a) Für 100 SFrs zahlen Sie 109,50 DM, wieviele Schweizer Franken (f) erhalten Sie für 250 DM (d)? Rechne:

$$250 \cdot \frac{109{,}5}{100} = 273{,}75\ (?);\quad 250 \cdot \frac{100}{109{,}5} \approx 228{,}31\ (!);$$

„Erinnerung": $f = d : 1{,}095$

b) Wie hieß doch das „einfachste" Weg-Zeit-Geschwindigkeitsgesetz? So:

$$t = s \cdot v?\ \text{oder}\ s = v \cdot t?\ \text{oder}\ v = s \cdot t?\ \text{oder}\ v = \frac{s}{t}?$$

Das Entdecken von Bildungsgesetzen oder Rekursionsformeln für Folgen bei Vorgabe einiger Folgenglieder erfolgt prinzipiell durch ein Probierverfahren, bei dem der ETR schnell entsprechende Vermutungen verifiziert bzw. falsifiziert. Zur Entdeckung von Funktionseigenschaften (Monotonie, Grenzwerte ...) berechne und beobachte:

- Sekantensteigungen bei kleiner werdender „Schrittlänge"
- $s_{n+1} = s_n \cdot q + a;\ s_1 = 1;\ a = 10$
  $q \in \{-2; -1{,}5; 0; 0{,}01; 0{,}1; 0{,}5; 1; 1{,}5\}$
- $w_{n+1} = 1 + \frac{n}{1 + w_n};\ w_1 = 1;\ n \in \{1, 2, 3, \ldots\}$
- $c_{n+1} = \cos c_n,\ c_1 = 0{,}74$
- $x_{n+1} = ((x_n^2 + a) : x_n) : 2;\ x_1 = 1,\ a \in \mathbb{R}^+$
- $d_{n+1} = \sqrt{\sqrt{a \cdot d_n}};\ d_1 = 1,\ a \in \mathbb{R}^+$
- $k_{n+1} = 2k_n - a \cdot k_n^2;\ k_1 \in \{0; 0{,}1; 0{,}2; 0{,}5; 0{,}9; 1; 1{,}2\},\ a \in \mathbb{R}^+$

Bei der Folge $\langle k_n \rangle$ wird z.B. sehr schnell klar, daß der „Fixpunkt" $\left(\frac{1}{a}\right)$ nicht für beliebige Anfangsglieder erreicht wird.

Es muß dem Schüler verdeutlich werden, daß

- eine Vermutung noch kein Nachweis (Beweis) ist,
- man sich täuschen kann,
- eine unvollständige Induktion (punktuelles Verifizieren) keine vollständige Induktion ist.

### 4.4 Algorithmisches Denken

Algorithmisches Denken zielt auf folgende Kenntnisse und Fertigkeiten:

- Kenntnis von Problemlösungs-Methoden (-Strategien, -Verfahren) und deren Darstellungsweise;
- Fähigkeit, diese (Algorithmen) auf konkrete Probleme anwenden zu können (Exemplifizieren);
- vorgegebene Algorithmen so zu ergänzen, daß sie auf eine umfassendere „Problemklasse" anwendbar sind (Generalisieren);
- leistungsgleiche Verfahren kritisch aufgrund vorgegebener oder zu entwickelnder Kriterien miteinander vergleichen zu können;
- neue Algorithmen selbsttätig zu entdecken.

Charakteristisch für algorithmisches Denken sind:

- rekursives Definieren und Arbeiten (Durchlaufen von Schleifen),
- logisches Denken auf verzweigten Wegen,
- induktives Schließen als „Argumentieren zur nächsten Stufe".

Im folgenden soll an wenigen Beispielen der Stellenwert des ETR in einem hierauf orientierten Mathematikunterricht der S I angedeutet werden. Dabei stehen solche Algorithmen im Vordergrund, die auch mit einfachen, nicht programmierbaren ETR ausführbar sind.

*Beispiele:*

- Teilen mit Rest:
  a) Division als wiederholte Subtraktion
  b) Bestimmung des ganzzahligen Anteils [x] für $x \in IR$
- EUKLIDISCHER Algorithmus für ggT (a, b) und Kettenbrüche
- Kleinstes gemeinsames Vielfaches kgV (a, b)
- Primzahlen und Primfaktorzerlegung
- Umwandlungen von Zeitangaben (Tage/Stunden/Minuten/Sekunden), Winkelmaßzahlen (Grad/Minute/Sekunde) oder Brüchen in Dezimalzahldarstellung und deren Umkehrung
- Bestimmung von mehr sicheren Ziffern als der ETR anzeigt

(z.B. $\frac{1}{47} = 0{,}\overline{021\ 276\ 595\ 744\ 680\ 851\ 063\ 829\ 787\ 234\ 042\ 553\ 191\ 489\ 361\ 7}$)

- Zahldarstellung in verschiedenen Stellenwertsystemen
- Rekursiv definierte Folgen

  Speziell: $S_{2n} = \sqrt{2-\sqrt{4-s_n^2}} = s_n : \sqrt{2+\sqrt{4-s_n^2}}$;

  $s_6 = 1 \Rightarrow n \cdot s_{2n} \rightarrow \pi$-Approximation
- Bestimmung von n-ten Wurzeln (Heron; $x_{n+1} = \sqrt{\sqrt{ax_n} \rightarrow \sqrt[3]{a}}$)
- Simulation von Wachstums- und Zerfallsprozessen
- Simulation von Ratensparverträgen
- Hornerschema
- „Probiermethoden" zur Lösung von (Un-)Gleichungen
- Intervallschachtelung
- Rundungs- und Fehlerabschätzungsmethoden
- Was muß man tun, um dasselbe zu bewirken wie ein ETR nach Betätigen z.B. der Tasten

  [|x|], [n!], [σ], [Int],

  [M+], [M + x²] bzw. [x ↔ y] ?

### Literatur


[1] *Bardy, P.:* Der Einsatz von Taschenrechnern im Mathematikunterricht berufsbildender Schulen – Aspekte und Beispiele, ZDM 3/78, S. 134 ff.

[2] *Engel, A.:* Elementarmathematik vom algorithmischen Standpunkt, Klett 1977.

[3] *Engel, A.:* The Role of Algorithms and Computers in Teaching Mathematics at School, 3. Int. Kongreß über Mathematikunterricht, Karlsruhe 1976

[4] *Frei, H. P.* e.a.: Nach dem Taschenrechner: Wie soll der ideale Schulcomputer aussehen? ZDM 4/78, S. 217–221.

[5] *Herget, W.* e.a.: Taschenrechner und Rechenstab im Mathematikunterricht – eine aktuelle Schüler- und Lehrerbefragung –, Praxis Mathematik, H. 7, Juli 78.

[6] *Jäger, J.:* Rechenhilfen sind Kulturgut der angewandten Mathematik, ZDM 3/78.

[7] *Meißner, H.:* Taschenrechner im Mathematikunterricht für 5- bis 12-jährige, ZDM 4/78.

[8] *Meißner, H.:* Welcher Taschenrechner ist am besten? Neue Unterrichtspraxis, 2/April 1977, Schroedel Hannover.

[9] *Müller, K. P.:* Lehreraus- und Fortbildung für den Taschenrechnereinsatz ZDM 3/78.

[10] Stellungnahme der GDM zum Einsatz von ETR im Mathematikunterricht, ZDM 2/78, S. 117.

[11] *Schmitt, H.:* Rechenfertigkeit und el. Taschenrechner in der Hauptschule, Westermanns pädagogische Beiträge 29, 1977, S. 115–119.

[12] *Wynands, A. u. U.:* Elektronische Taschenrechner in der Schule – Ein Arbeits- und Aufgabenbuch für Lehrer und Schüler – Vieweg 1978.

[13] *Wynands, A.:* Elektronische Taschenrechner in der Hauptschule – Ziele und Erfahrungen, MU 1/78.

[14] *Wynands, A.:* Ergebnisse einer Schüler- und Lehrerbefragung über „Elektronische Taschenrechner in der Schule", ZDM 1/78.

[15] *Wynands, A.:* Zur fachdidaktischen Komponente des elektronischen Taschenrechners, ZDM 3/78.

Mit dem Thema „Taschenrechner in der Schule" brauchen wir uns nicht mehr auseinanderzusetzen, da Chancengleichheit durch Preisverfall gewährleistet wurde, die Kultusministerien inzwischen ihre Erlasse pro Rechner und kontra Rechenschieber ändern und selbst verängstigte Eltern oder vorsichtige Pädagogen den Sinn und Zweck des Taschenrechners als Rechenhilfe begriffen haben und bei richtigem Einsatz keine schleichende Schülerverdummung mehr befürchten. Damit sieht sich der Pädagoge jedoch vor die schwierige Aufgabe gestellt, aus einer Vielzahl von Herstellern und Geräten das Optimale für seine Schüler und damit für die didaktisch beste Gestaltung seines Unterrichts herauszusuchen. Der nachfolgende Beitrag soll kurz darstellen, welches die Hauptbestandteile des Taschenrechners sind, welche Verarbeitungslogiken es gibt, um darauf aufbauend Kriterien für die allgemeine Auswahl von Schulrechnern zu erstellen.

D. Nenner

# Intelligente Taschenrechner mit AOS

## Auswahlkriterien für Schulrechner

### 1 Hauptbestandteile eines Taschenrechners

Das Äußere eines Taschenrechners (TR) wird geprägt vom Tastenfeld, der Anzeige und dem Gehäuse selbst. Dabei dient das Tastenfeld zur Eingabe von Zahlen, Operationen und Funktionen; die Anzeige macht eingegebene Ziffern und errechnete Daten sichtbar, und beim Gehäuse zeichnet sich ein Trend zur Flachbauweise ab durch die Verwendung von *Flüssigkristallanzeigen.*

Geht man davon aus, daß ein Taschenrechner ein Computer im Kleinformat ist, dann hat er folgende drei Hauptfunktionen: die Eingabe, die interne Verarbeitung der Daten und die Ausgabe. Auf dem nachfolgenden Blockschaltbild sind die wichtigsten TR-Bestandteile, besonders auch die der internen Logik, graphisch dargestellt.

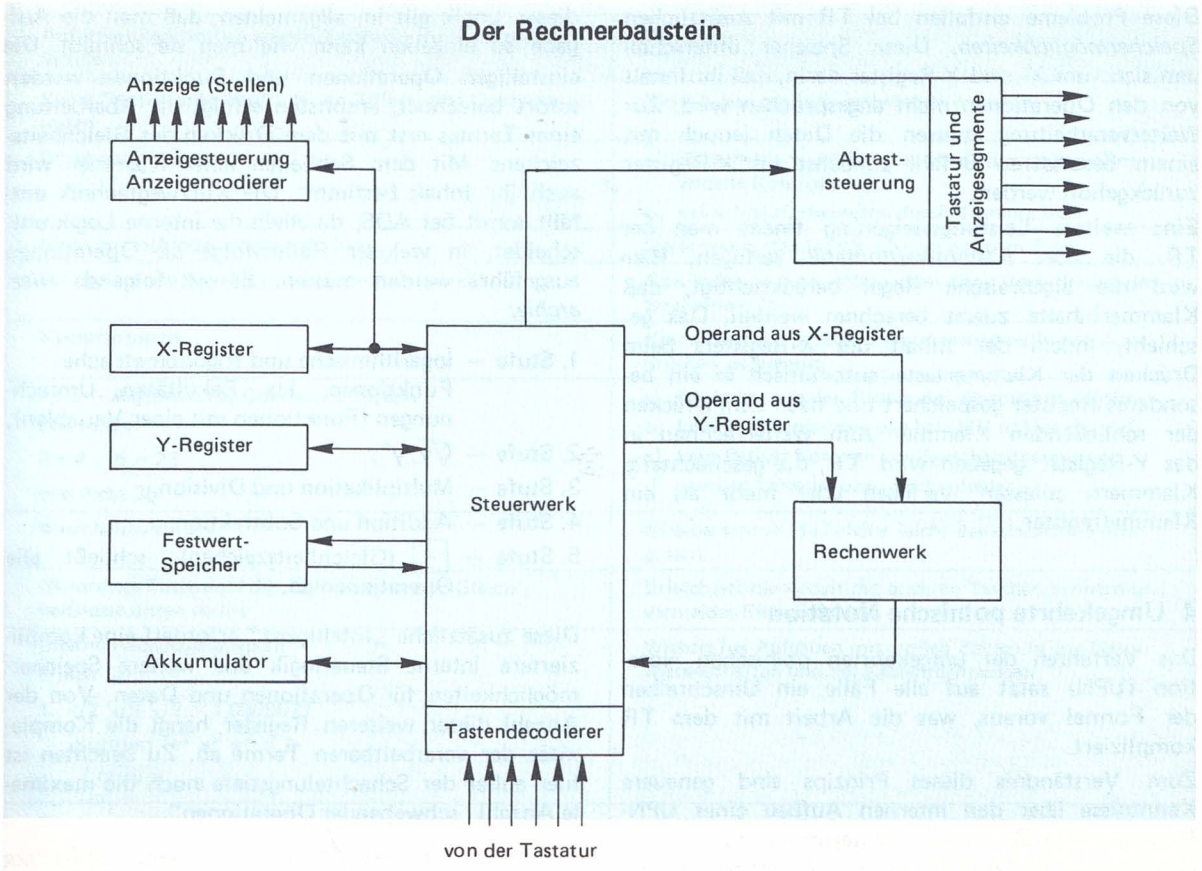
Die Eingaben – oder die von der Tastatur kommenden Leitungsbahnen – führen zum *Tastendecodierer.* Die interne Verarbeitung der Daten wird im *Rechenwerk* unter der Kontrolle des *Steuerwerkes* vorgenommen. Im Zusammenhang mit arithmetischen Operationen, die hier auf Binärzahlen zurückgeführt werden, sind Verknüpfungen in den *Registern* notwendig.

Ergebnisse bzw. Eingaben von der Tastatur werden über den *Anzeigencodierer* zum *X*-Register und zur Anzeige geleitet, die die Ausgabeeinheit des TR darstellt. Beide Codierer, die Register, Speicher, der Akkumulator und das Rechenwerk werden funktionell durch das Steuerwerk überwacht.

### 2 Taschenrechner-Logiken

Alle TR sind in der Lage, ein- und zweistellige Operationen und Funktionen auszuführen, deren Operanden jeweils im *X*-Register bzw. im *X*- und *Y*-Register stehen. Bei den neueren Rechnermodellen unterscheidet man nur noch zwischen den *algebraischen Logiken* und der *umgekehrten polnischen Notation* (UPN).

Es ist jedoch nicht angebracht von nur einer algebraischen Notation zu sprechen, denn es gibt hier sehr verschiedene Varianten (ohne Klammerautomatik, mit einfachen und geschachtelten Klammern, mit begrenzter *Hierarchie* bis hin zum so-




genannten AOS, dem *Algebraischen Operations System*). Diese Vielfalt wirkt sich stark auf die Handhabung der Rechner aus. Anders ist das bei TR mit umgekehrter polnischer Notation, die vorwiegend im Bereich der Datenverarbeitung Verwendung findet. Im Schulbereich kann auf die allgemeinen Regeln der Algebra nicht verzichtet werden.

Im folgenden werden die wesentlichen Gemeinsamkeiten und Unterscheidungsmerkmale der einfachen algebraischen Notation dargestellt, eine Abgrenzung gegenüber der umgekehrten polnischen Notation vorgenommen und das Algebraische Operations System erklärt.

## 3 Einfache algebraische Logiken

Die meisten TR verfügen heute über die *Kurzwegtechnik* was bedeutet, daß nicht nur die Gleichheitstaste, sondern auch die Eingabe eines neuen Operationszeichens die Ausführung der zuletzt eingegebenen Operation bewirkt. Das Ergebnis dient als erster Operand für die folgende Operation. Hiermit wird zwar die Anzahl der Tastendrucke im Laufe einer Rechnung stark verringert, doch werden in vielen Fällen Umstellungen in den Formeln unumgänglich, z.B. bei der Aufgabe:

$10 - 2 \times 3 + 8 = 32$ (falsch);

erst nach Umstellung auf:

$2 +/- \times 3 + 10 + 8 = 12$

erhält man das richtige Ergebnis.

Erweitert man die Aufgabe noch durch eine Division, dann ist man bei einfachen TR-Typen gezwungen, die Rechnung zu unterbrechen, Zwischenergebnisse zu notieren und wieder neu einzugeben.

Diese Probleme entfallen bei TR mit zusätzlichen *Speichermöglichkeiten.* Diese Speicher unterscheiden sich vom *X*- und *Y*-Register darin, daß ihr Inhalt von den Operationen nicht angesprochen wird. Zur Weiterverarbeitung müssen die Daten jedoch mit einem besonderen Befehl zunächst ins *X*-Register zurückgeholt werden.

Eine weitere Leistungssteigerung findet man bei TR, die über *Klammerautomatik* verfügen. Hier wird die algebraische Regel berücksichtigt, daß Klammerinhalte zuerst berechnet werden. Das geschieht, indem der Inhalt des *X*-Registers beim Drücken der Klammertaste automatisch in ein besonderes Register gespeichert und nach dem Drücken der schließenden Klammer zum Weiterrechnen in das *Y*-Register gegeben wird. TR, die geschachtelte Klammern zulassen, verfügen über mehr als ein Klammerregister.

## 4 Umgekehrte polnische Notation

Das Verfahren der umgekehrten polnischen Notation (UPN) setzt auf alle Fälle ein Umschreiben der Formel voraus, was die Arbeit mit dem TR kompliziert.

Zum Verständnis dieses Prinzips sind genauere Kenntnisse über den internen Aufbau eines UPN-Rechners unerläßlich. Im Unterschied zu dem algebraischen Verfahren werden die arithmetischen Operationen in einem *Rechenregister – Stapel,* genannt „Stack", ausgeführt, der aus vier übereinander angeordneten „Stackregistern *X, Y, Z, T*" besteht, die die zu verarbeitenden Daten aufnehmen. Da die Eingabe einer Operation deren sofortige Ausführung mit den in *X*- und *Y*-Register befindlichen Operanden bewirkt, muß der Anwender sich laufend über die Inhalte des Stacks im klaren sein. Er muß berücksichtigen, in welcher Weise die Befehle die Anordnung der Daten beeinflussen, damit er die Aufgabe nicht falsch berechnet. Außerdem gibt es exakt einzuhaltende Eingabevorschriften, bei denen die internen Voraussetzungen für die Lösung berücksichtigt werden müssen.

## 5 Das algebraische Operations System (AOS)

Die höchste Leistungsfähigkeit eines TR wird mit dem Algebraischen Operations System erreicht. Hier wird nicht nur die Klammerhierarchie berücksichtigt, sondern auch die Hierarchie unter den Operationen und Funktionen. Erst bei TR mit dieser Logik gilt im allgemeinen, daß man die Aufgabe so eingeben kann wie man sie schreibt. Die einstelligen Operationen und Funktionen werden sofort berechnet, ansonsten erfolgt die Abarbeitung eines Termes erst mit dem Drücken des Gleichheitszeichens. Mit dem Schließen einer Klammer wird auch ihr Inhalt bestimmt. Die Kurzwegtechnik entfällt somit bei AOS, da allein die interne Logik entscheidet, in welcher Reihenfolge die Operationen ausgeführt werden müssen. Es gilt folgende *Hierarchie:*

1. Stufe – logarithmische und trigonometrische Funktionen, 1/x, Fakultäten, Umrechnungen (Funktionen mit einer Variablen),
2. Stufe – $\sqrt[x]{y}$, $y^x$,
3. Stufe – Multiplikation und Division
4. Stufe – Addition und Subtraktion
5. Stufe – [=] (Gleichheitszeichen), schließt alle Operationen ab.

Diese zusätzliche „Intelligenz" erfordert eine kompliziertere interne Steuerlogik und weitere Speichermöglichkeiten für Operationen und Daten. Von der Anzahl dieser weiteren Register hängt die Komplexität der verarbeitbaren Terme ab. Zu beachten ist hier außer der Schachtelungstiefe noch die maximale Anzahl „schwebender Operationen".

Allgemein ist festzustellen, daß eine einfachere Eingabe ein komplizierteres Rechnersystem voraussetzt. Jedoch nur durch die Kombinationen mit dem Algebraischen Operations System (AOS) nimmt einem ein TR Überlegungen zur Termumformung ab. Das bedeutet:

- weniger Rechenfehler,
- weniger Eingabefehler,
- kein lästiges Notieren von Zwischenergebnissen,
- der Rechner arbeitet als Hilfsmittel des Anwenders und wird nicht Teil des zu lösenden Problems.

## 6 Anforderungskriterien an Schulrechner

Nimmt man die im vorhergegangenen Text beschriebenen faktischen Gegebenheiten über Aufbau und Rechnerlogiken von TR, so läßt sich eine Liste von Kriterien erstellen, die, bei Berücksichtigung durch den Lehrer, den TR als ein methodisch sinnvolles Hilfsmittel im Unterricht Verwendung finden lassen.

| Anforderungskriterien an den Taschenrechner für die Schule | Begründung |
|---|---|
| Keine Doppelfunktionen auf den Ziffern- und Operationstasten | Tests haben große Eingabefehler erwiesen |
| Druckpunkttasten | a) Rückmeldung der Eingabe über Fingerspitze ohne visuelle Kontrolle<br>b) keine Mehrfacheingabe durch Tastenprellen |
| Nicht zu flach oder spielerisch dünn | Gerät wenig stand- oder lagefest auf dem Tisch |
| Schrägliegende Anzeige mit weitem Sichtwinkel | Gewährleistet gute Ablesbarkeit ohne Veränderung der Sitzhaltung |
| Klammertasten | Keine laufende Umstellung von mathematischen Ausdrücken notwendig |
| AOS – Algebraisches Operations System<br>Prüfen Sie:<br>$3 + 4 \times 5 = 23$<br>und nicht 35 | a) Rechnet nach den Regeln der allgemeinen Algebra<br>b) kein Termumformen wie bei UPN notwendig<br>c) kein lästiges Notieren vonZwischenergebnissen<br>d) weniger Eingabe- bzw. Rechenfehler |
| Ausschaltautomatik | Stromersparnis, da Schüler leicht das Abschalten vergessen. |
| Genormtes Tastenfeld der Zifferntasten und Gleichheitstaste unten rechts | Erleichtert die Arbeit mit anderen Taschenrechnern und vermeidet Eingabefehler. |
| Interne Rechengenauigkeit<br>Prüfen Sie:<br>1. $\sqrt{2}\ \sqrt{2}\ \sqrt{2} = 1{,}0905077$<br>quadriert $x^2\ x^2\ x^2 = 2$<br>2. $\sqrt{e^{(2\,LN\,3)}} = 3$ | Wichtig bei Aufgaben mit großen Zahlen in den Naturwissenschaften und bei Kettenrechnungen |
| Direkte Berechnung von $\sqrt[x]{y}$ und $x^2$ muß über Tasten möglich sein. | Häufig auftretende Berechnungen, die bei Umstellungen leicht zu Fehlern führen. |
| Mindestens ein Speicher | Zur Speicherung weiterer Werte oder Konstanten |
| Fehleranzeige im Display | Zur Warnung vor unlogischen Eingaben oder bei Überforderung des TR |
| Exponentialdarstellung | Unverzichtbar bei Aufgaben mit hoher Genauigkeit oder bei großen Zahlen. |
| Automatische Löschung der Konstante | Fehlrechnungen durch unbewußte Konstantenbenutzung |
| Batteriegeräte sollten umrüstbar sein auf Akku- oder Netzbetrieb (Ausnahme LCD) | Ersparnis laufenden Batteriekaufs |
| Wenig interne Bauelemente (Öffnen Sie die Geräte zum Vergleich: je weniger Bauteile sie vorfinden, um so höher der Integrationsgrad). | Hoher Integrationsgrad bedeutet geringe Anzahl von Fehlerquellen. |

Unter Zuhilfenahme der vorgenannten Anforderungskriterien an einen Schulrechner, wird der Erfolg nicht ausbleiben:

- Anhaltende Motivation von Schülern, d.h. mehr Spaß am Mathematikunterricht bei wesentlicher Verringerung der Rechenfehler
- leichtere Einführung in mathematische Zusammenhänge durch den hohen Aufforderungscharakter eines Taschenrechners, denn er erzwingt eine interaktive Arbeitsweise mit sofortiger Rückmeldung
- Konzentration auf mathematisches Verständnis und Wege zur Problemlösung, d.h. Kreativität hat Vorrang vor Routineberechnungen.

Unter diesen Prämissen ist ein Taschenrechner in der Lage, dem Benutzer soviel Arbeit wie möglich abzunehmen, womit er zur Lösung der Probleme beitragen kann und nicht selbst ein Teil des Problemes wird.

Man kann das Programmieren lernen und dann DV-Maschinen sinnvoll nutzen, ohne etwas über die technischen und logischen Abläufe im Rechner zu wissen. Andererseits läßt sich das Erlernen der Programmierung aber auch als Werkzeug einsetzen, mit dem Funktionsweise und Struktur von elektronischen Rechnern verständlich werden. Der zweite Weg wird hier beschrieben. Hervorzuheben ist die günstigere Beurteilung von Flußdiagrammen gegenüber Struktogrammen. Schneider beschreibt in seinem Aufsatz „Grafische Darstellung von Programmabläufen" im Mikrocomputer-Teil dieses Jahrbuches weitere Erfahrungen dazu.

H. Nahrstedt

# Programmierbare Taschenrechner und Mikrocomputer in der Erwachsenenbildung

## Erfahrungen an der VHS Hamm

Die Volkshochschulen bemühen sich von Jahr zu Jahr, ihr Angebot weiter auszubauen. So wurde erstmalig im Herbst 77 an der VHS Hamm der Kursus „Einführung in das Programmieren von Taschenrechnern" eingeführt. Im Herbst 78 kam noch ein Kursus „Mikrocomputer" hinzu. Die dabei gemachten Erfahrungen haben unser Konzept ständig verändert. Mit einem Kursus „Programmieren in BASIC" hoffen wir unsere Palette ab Herbst 79 abzurunden.

Wegen der oftmals unterschiedlichen oder falschen Erwartungen, mit denen unsere Teilnehmer zu uns kommen, möchte ich an dieser Stelle die Gelegenheit ergreifen und die Vorstellungen über diese Kurse korrigieren. Vielleicht ist dies aber auch ein Anreiz für manchen Leser, sich auch für anderen Orts bestehende Kurse zu interessieren.

Da ist zunächst der Hörerkreis zu nennen, der in dieser Komplexität und mit den Motivationen wohl in keiner anderen Schulform auftritt. Angefangen von einer Hausfrau, die ihr Budget gern mit einem Taschenrechner überprüfen möchte und deren Küchengeräte bereits per Mikrocomputer gesteuert werden, über den Schüler, der einen programmierbaren Taschenrechner im Unterricht einsetzt und in einer Arbeitsgemeinschaft an einer Tischrechenanlage lernt, bis hin zum Ingenieur, dessen ständige Berechnungen ein programmierbarer Taschenrechner erleichtern kann und dessen geplante Maschinen durch einen Mikrocomputer gesteuert operieren sollen, ist alles vertreten. Dieser Hörerkreis verlangt eine Normierung, d.h. es muß vor dem eigentlichen Thema soviel Grundwissen vermittelt werden, daß alle „die gleiche Sprache sprechen".

Sinn und Ziel dieser Kurse ist es allerdings nicht, ein fundiertes Grundwissen zu vermitteln, sondern vielmehr Einblick in ein neues Wissensgebiet zu geben und einen Grundstein zu legen für eine spätere, gezielte Fortbildung in Schulungskursen oder auf autodidaktischem Wege. Es wird gerade dem Anfänger ermöglicht, die besonders auf dem Computermarkt bestehende Informationsflut zu übersehen und die für ihn interessanten Informationen richtig einzuordnen. Dabei können oftmals falsche Meinungen revidiert und fälschlich angefangene Bildungswege umgelenkt werden.

Den Lerninhalt beider Kurse kann man grob in zwei Hälften teilen. Der 1. Teil befaßt sich mit der Betrachtung allgemeiner *Strukturen und Regeln*, und im

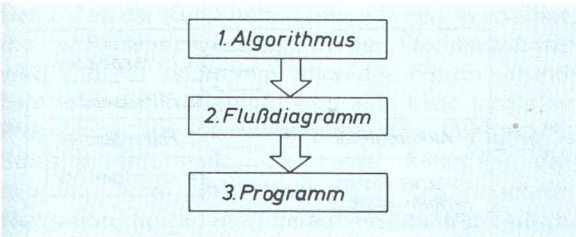



**Bild 1** Der Weg von einer Idee zum Programm

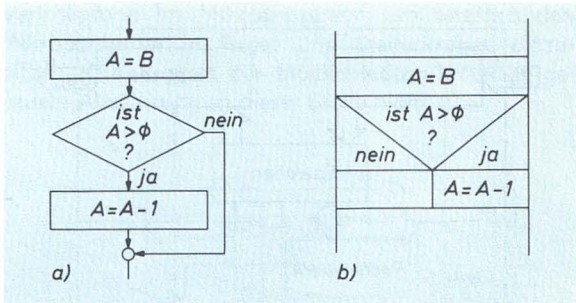



**Bild 2** Vergleich Flußdiagramm (a) und Struktogramm an einer bedingten Verzweigung (b)

2. Teil erfolgt dann die Hinwendung zu konkreten *Computertypen*. Während man den 1. Teil wiederum konkret nach *Software* (Programmierung) und *Hardware* (Computerstrukturen) trennen kann, setzt im 2. Teil die Programmierung auch Kenntnisse der vorhandenen **Rechnerstruktur** voraus.

Es wird nachfolgend die Kursusstruktur weiter detailliert. Der Softwareanteil des 1. Kursusteiles beginnt mit der Definition des Begriffs **Algorithmus** und der praktischen Demonstration an Beispielen. Eins der klassischen Beispiele ist der Euklidische Algorithmus zum Finden des größten gemeinsamen Teilers (ggT) zweier natürlicher Zahlen. Hier werden Grundlagen zur **linearen Programmierung** gelegt, Einblicke in den zeitlichen Ablauf einer Problemlösung gegeben und Begriffsvorstellungen darüber geschaffen, welche Probleme durch Computer lösbar sind.

Die Übersetzung eines Algorithmus ins *Flußdiagramm* ist der zweite Schritt auf dem Weg zum fertigen Programm, wie es **Bild 1** graphisch wiedergibt. Dabei habe ich immer wieder festgestellt, daß das Aufstellen eines Flußdiagramms allen Teilnehmern am leichtesten fällt. Es liegt wohl am graphischen Anteil, denn man hat sozusagen „auf einen Blick" alle Zusammenhänge erfaßt. Die alternativen **Struktogramme** sind hier weniger wirksam. Außerdem sind sie für einfache Programmiersprachen auch weniger geeignet. **Bild 2** verdeutlicht an einem Beispiel beide Methoden. Nach mehreren Übungen wird in der Regel die schriftliche Formulierung eines Algorithmus fallengelassen und die Teilnehmer beginnen gleich mit dem Flußdiagramm. Allgemein wird immer die Übersichtlichkeit bei der Entwicklung im Flußdiagramm hervorgehoben. Der dritte Schritt, die Übersetzung des Flußdiagramms in das jeweilige *Rechnerprogramm* fällt dann am leichtesten, wenn zuvor die Rechnersprachelemente an Beispielen erklärt werden. Hier ist es für die spätere Übersichtlichkeit eines Programms wichtig, daß man gewisse Grundregeln und Gesetzmäßigkeiten einführt. Zum Rüstzeug für das Programmieren gehören weiterhin der Aufbau von **bedingten Sprüngen** und *Unterprogrammen*, sowie das Prinzip der **indirekten und relativen Adressierung**. Eventuell muß man bei Rechnern, die über bestimmte oftmals notwendige Grundfunktionen nicht verfügen, Hilfsprogramme einführen.

Bei der *Hardware* kann man für alle Gebiete von einer Grundstruktur, wie **Bild 3** sie zeigt, ausgehen. Diese wird dann in den einzelnen Richtungen konkretisiert. Bei den programmierbaren Taschenrechnern ist es insbesondere die Betrachtung der *Arbeitsspeichereinheit* (ASE) **Bild 4.**

**Rechnerstruktur:** Logischer Aufbau eines DV-Systems.

**Algorithmus:** Rechenvorschrift, festgelegtes Schema für einen Lösungsweg.

**Lineare Programmierung:** Entwickeln und Schreiben eines Programms in der logischen Folge, die dem Ablauf des zu programmierenden Vorgangs entspricht.

**Struktogramm:** Grafische Darstellungsmöglichkeit für Programmabläufe, wobei nicht die Reihenfolge im Vordergrund steht (wie beim Flußdiagramm), sondern die Programmstruktur (s. *Schneider*: Grafische Darstellung von Programmabläufen).

**Bedingter Sprung:** Verzweigung in einem Programmablauf, die durch Abfrage einer vorgegebenen Bedingung zustande kommt.

**Indirekte Adressierung:** Während „normalerweise" im Adreßteil eines Programmbefehls direkt die Speicherzelle angegeben ist, in der die gewünschten Daten stehen (*direkte Adressierung*), wird hier eine Speicherzelle genannt, die erst die Adresse enthält.

**Relative Adressierung:** Die Zieladresse, die den nächsten auszuführenden Befehl oder gewünschte Daten enthält, wird hierbei als Abstand, also relativ zu einer festgelegten Adresse angegeben.

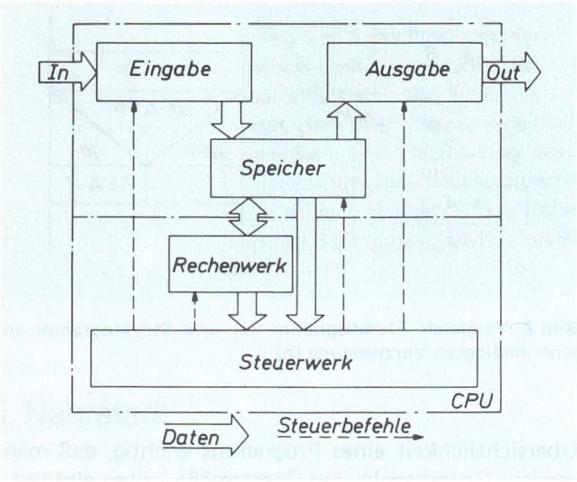



**Bild 3** Grundstruktur eines Computers (In: Eingabe; Out: Ausgabe)

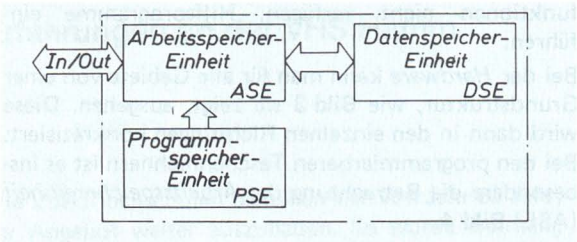



**Bild 4** Gliederung der Speichereinheit von programmierten Taschenrechnern

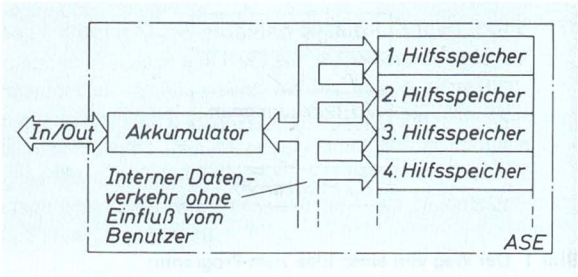



**Bild 5** Arbeitsspeichereinheit (ASE) bei algebraischer Notation

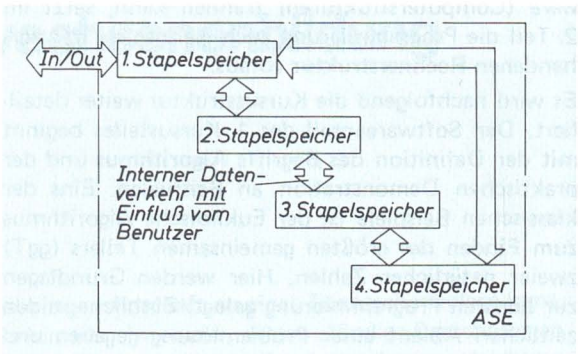



**Bild 6** Arbeitsspeichereinheit (ASE) bei umgekehrter polnischer Notation

Je nach der *Notation des programmierbaren Taschenrechners* handelt es sich bei der Arbeitsspeichereinheit (ASE) um einen **Akkumulator** mit zusätzlichen Hilfsregistern bei **algebraischer Notation (Bild 5)** oder um einen Stapelspeicher **(Stack)** bei **umgekehrter polnischer Notation (Bild 6).**

Bei den *Mikrocomputern* kann man zu einem einfachen Blockbild wie **Bild 7** es zeigt übergehen, bei dem man danach bereits sehr konkret auf einen bestimmten *Mikroprozessor* zu sprechen kommen muß. Eine Alternative ist die Einführung eines hypothetischen Mikroprozessors mit minimaler Ausstattung, den man je nach Wissensstand weiter ausbaut, bis seine Struktur dem vorliegenden Mikroprozessor ähnelt. Danach kann man zum Computer wechseln.

**CPU:** *Central Processing Unit*; Zentraleinheit einer EDV-Anlage.

**Akkumulator:** Das wichtigste Arbeitsregister in der CPU, in dem die meisten Operationen ausgeführt werden.

**Stack:** Spezieller Speicherbereich (*Stapelspeicher*), der in der Regel durch die CPU automatisch genutzt wird (z.B. bei Programmunterbrechungen oder Unterprogrammaufrufen).

**Algebraische Notation:** Von den meisten Taschenrechnern genutztes Verfahren zur Eingabe und Verarbeitung, z.B. a + b =: Eingabe in dieser Reihenfolge.

**Umgekehrte Polnische Notation:** Eingabemethode, die ohne Klammersetzung auskommt und „Punktrechnung" vor „Strichrechnung" richtig behandelt.

Der 2. Teil der Kurse befaßt sich, wie bereits erwähnt, mit der Rechneranwendung. Bei den Taschenrechnern wird von mir neuerdings folgendes Prinzip gehandhabt: An die Teilnehmer wird eine Liste möglicher Programme aus den Wissensgebieten Mathematik, Statistik, Informatik, Elektronik, Finanzen, Vermessung, Astronomie, Mechanik, Ingenieurtechnik, Navigation, Spiele, usw. herausgegeben. Die Teilnehmer haben dann die Möglichkeit, von ihnen gewünschten Programme anzukreuzen oder eigene Vorschläge zu unterbreiten, die dann mit in eine neue Übersicht aufgenommen werden. Außerdem wird der eigene, vorhandene Taschenrechner angegeben. Die gewünschten Programme werden so später im Kurs auf diesen zugeschnitten, und der Teilnehmer hat damit optimale Anwendungsbeispiele. Darüber hinaus wird den Teilnehmern ein Einblick in die verschiedenen Notationen und Programmiermöglichkeiten der auf dem Markt befindlichen Rechner vermittelt.

Beim Mikrocomputer-Kursus gehen wir von einem einzigen Typ aus, dem Prozessor 6502, der sich wiederum im PET 2001 befindet. Hier ist die besondere Vielfalt von Anwendungsmöglichkeiten wichtig. Da gehört zunächst das Benutzen vorhandener **Peripheriegeräte** zur Einführung. Alsdann betrachten wir die Ein-Ausgabe analoger und digitaler Werte und ihre Verknüpfung im Mikroprozessor, um letztlich den Mikrocomputer in Regel- und Steuerkreisen einzusetzen. Die Vielfalt der Möglichkeiten läßt eine genauere Ausführung an dieser Stelle nicht zu.

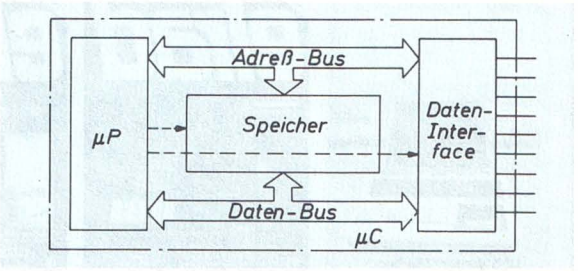



**Bild 7** Blockbild eines einfachen Mikrocomputers

**Peripherie:** Alle Bauteile und Geräte einer EDV-Anlage, die außerhalb der CPU angeordnet sind (z.B. Bedienungsgeräte, externe Speicher).

**Interface:** Schnittstelle zwischen CPU und Peripherie.

**Einteilung von Rechnerklassen**

- *Großcomputer* als aufwendigste Spezialrechner für Technik, Wissenschaft und Rechenzentren (z.B. Kerntechnik, Raumfahrt, Zentralregister);
- *Mittlere Anlagen* für den kommerziellen Bereich („Mittlere Datentechnik", MDT);
- *Prozeßrechner* zur Meßdatenverarbeitung in Industrie und Wissenschaft sowie zur Überwachung, Steuerung und Regelung technischer Produktionsprozesse;
- *Minicomputer* für die „Dezentrale Datenverarbeitung" und als Prozeßrechner;
- *Mikrocomputer* für die noch weiter dezentralisierte Datenverarbeitung, Datenvorverarbeitung in „intelligenten Terminals" sowie als anpassungsfähiges Mittel für die Meßtechnik und für einfachere Steuerungs- und Überwachungsaufgaben;
- *Programmierbare Tisch- und Taschenrechner* als bequeme und preiswerte Hilfsmittel in unzähligen Anwendungsfällen.

# Kennen Sie eigentlich schon . . .

**elrad** magazin für elektronik

**Ein Magazin aus dem**
**Verlag Heinz Heise Hannover**
Kommanditgesellschaft
Abt. III
**Postfach 27 46 · 3000 Hannover 1**

## Das Magazin für den Hobby-Elektroniker

— ·ob Anfänger oder Spezialist —
— Audio — Meßtechnik — Amateurfunk
— Digitaltechnik — Mikrocomputer

### Elrad bringt

Übersichtsartikel über neueste Entwicklungen auf dem Elektronik-Sektor. Verständlich geschrieben. Ohne theoretischen Ballast.

Erprobte Bauanleitungen. Mit Druckplatinenvorlagen und Bestückungsplänen.

'Tech-tips'. Schaltungen, Kniffe, Tricks. Aus der Praxis — für die Praxis.

In jeder Ausgabe auf zwei Seiten englischsprachige Originalartikel. Aus Elektrotechnik und Elektronik. Mit deutschen Erläuterungen. Aussprachebezeichnungen in internationaler Lautschrift. Englisch-deutsches Fachvokabular. Damit Elrad-Leser ihre Englisch-Kenntnisse vervollständigen können.

Fortsetzungsserien über interessante Themen. Zum Beispiel: Aktive Filter, Op-Amps, Timerschaltungen. Mit Kochrezepten zum schnellen Dimensionieren.

Ab April 1979 neu: Computing Today im Innenteil von elrad. Mit ausführlichen Tests von Home-Computern, Mikroprozessor-Systemen, Software und vieles mehr, was den Hobby-Computeristen interessiert.

Echte Buchbesprechungen. Von unabhängigen Experten. Von Werken aus allen Fachverlagen.

### Was wurde bisher gebracht?

. . . Graphic Equaliser. Compander. Frequenz-Shifter. Audio-Spektrum-Analysator. CCD-Phaser. Musik-Synthesiser. Polyphasen SSB Exciter. 2m PAs. Morse-Tutor. Kurzwellen-Audion. Impedanz-Meßgerät. Digital-Thermometer. Phasen-Meßgerät. Funktionsgenerator. Transistor- und Dioden-Tester. NF-Millivoltmeter. PET-Bericht. CUTS Cassetten-Interface. Aquarium-Thermostat. Metall-Suchgerät. Annäherungsschalter. Digitaler Drehzahlmesser. Digital-Trainer. Inhalt eines PROMs — sichtbar gemacht. Verschlußzeit-Timer. Entwurf von Oszillatoren. Transmission-Line Lautsprecher. Drehzahlmesser für Modellflugzeuge. Elrad Mikrocomputer EHC-80 . . . und vieles mehr.

Seit April 1979 mit
**Computing Today** im Innenteil

Folgende Beiträge aus dem Bereich Mikrocomputer sind bisher erschienen:

1. **Bauanleitungen:** CUTS Cassetten-Interface. Mikrocomputersystem Delphin EHC 80. Inhalt eines PROMs sichtbar gemacht. Und als Leckerbissen: Elrad Triton Personal Computer.
2. **Berichte, Tests:** Nascom 1. Commodore PET. Heathkit-System H 8. TRS 80 auf dem Prüfstand. 'Das Baby von Heathkit' ET 3400.
3. **Allgemeine Beiträge:** Der Mikroprozessor nahegebracht (Serie). Speichersysteme mit Mikroprozessoren. Höhere Programmiersprachen. Adressierungsarten bei Mikroprozessoren. $\mu$P-Biographie. BASIC-Programmier-Kurs.

. . . und es geht Monat für Monat weiter mit interessanten Beiträgen.

. . . sind das nicht Gründe genug, Elrad-Leser zu werden?

## Elrad Special 1

**Das erste Sonderheft mit den beliebtesten Bauanleitungen und Beiträgen aus den bis jetzt erschienenen Heften.**

**Inhalt**

Musik-Synthesiser, Hammerwurf, Aktive Filter, Graphic-Equaliser, Digital-Thermometer, Schwipspegelmesser, Metronom, Frequenz-Shifter, CCD-Phaser, IC-Test- und Experimentiergerät, Audio-Spektrum-Analysator, Entwurf von Oszillatoren, Morse-Tutor, Rauscht Ihr Recorder?, Inhalt eines PROMs, Transistor- und Dioden-Tester, Audio-Oszillator, Funktionsgenerator, Digitaltrainer Digimax, Verschlußzeit-Timer, Digitaler Drehzahlmesser, Aquarium-Thermostat, Morse-Piepmatz.

**Ab sofort erhältlich! 128 Seiten!**
**Preis: DM 9,80 + 1,50 Versandspesen.**

Der Elrad-Versand liefert zu diesem Preis gegen Vorauszahlung auf Postscheckkonto Hannover 407 99-300 oder beiliegenden Verrechnungsscheck.

**Bitte beziehen Sie sich auf diese Anzeige und fordern Sie ein kostenloses Probeheft und das Inhaltsverzeichnis aller bisher erschienenen Elrad-Ausgaben an!**

# Mikrocomputer

## Inhaltsübersicht


Einführung . . . . . 54

*Harald Schumny*
Mikrocomputer – Grundbegriffe und Abgrenzungen . . . . . 55

*Helmut v. Falser*
Planen – Lernen – Gestalten
Das Programm des individuellen Dialogrechners . . . . . 61

*Harald Schumny*
Personal Computer . . . . . 67

*Werner Handke*
Mikroprozessoren – Hilfsmittel zur Programmentwicklung und Testmethoden . . . . . 69

*Harald Schumny*
Schnittstellen und Busanordnungen . . . . 79

*Peter Welzel*
Die Adressierung in Mikrocomputer-Systemen . . . . . 86

*Peter Welzel*
Interrupts in Mikroprozessor-Systemen . . 95

*Wolfgang Schneider*
Grafische Darstellung von Programmabläufen . . . . . 102

*Wolfgang Schneider*
Dialogsprache BASIC . . . . . 108

*Martin Hund*
Grundausbildung Mikroprozessor und Mikrocomputer – Eine experimentelle Einführung in Struktur und Arbeitsweise 121

*Max Gottlob*
Das TI-Lern- und -Lehrsystem TM990/189 . . . . . 126

*Roland Löhr*
Impulserzeugung mit dem R6522 VIA (Versatile Interface Adapter) . . . . . 130

*Bernd Strößner*
ALPHA 1 und BETA 8 – Ein Mikrocomputersystem aus Berlin . . . . . 134

*Hans-Joachim Stuckenberg*
Bipolare, schnelle Mikroprozessorsysteme . . . . . 138

*Rolf Krüger* und *Paul Jürgen Dickers*
Schnelle Mathematik für den TMS 9900 . 146

*Rolf Krüger* und *Paul Jürgen Dickers*
Mathematik-Assembler für Mikroprozessoren . . . . . 149

# Einführung

Unter einem Mikrocomputer verstehen wir die Zusammenschaltung eines Mikroprozessors mit Speichern und Bausteinen zur Ein- und Ausgabe von Daten und Programmbefehlen. Während man beim Kauf eines Taschen- oder Tischrechners immer ein vollständiges und sofort funktionsfähiges Gerät erwirbt, kann die Anschaffung eines Mikrocomputersystems auf sehr unterschiedliche Weise verlaufen.

**Möglichkeit 1:** Es wird ein Komplettsystem gekauft, z.B. der *Heimcomputer* PET für 2000 Mark oder ein sogenanntes *Entwicklungssystem* für vielleicht 50 000 Mark. In beiden Grenzfällen kann man in der Regel nach Einschalten des Gerätes sofort in beispielsweise der Programmiersprache BASIC arbeiten. Ein weiterer Ausbau dieser Komplettsysteme ist eigentlich immer vorgesehen.

**Möglichkeit 2:** Die Entscheidung fällt auf ein *Lernsystem.* Ein typisches Beispiel dafür ist der ständig billiger werdende KIM. Nach Anschließen eines Netzteiles kann man auch damit in der Regel sofort arbeiten, allerdings nur in der hexadezimal codierten Maschinensprache.

**Möglichkeit 3:** Falls unbeschränkte Flexibilität und Ausbaufähigkeit gefordert sind, wird die Anschaffung eines modularen Systems angemessen sein, das aus einer CPU-Karte, aus Speicher- und Ein-Ausgabekarten und diversen anderen Elementen bestehen kann. Dazu ist dann ein Gehäuse mit Netzteil und zentralen Verbindungsleitungen *(Systembus)* nötig. Je nach Anforderung kann auf Programmierung in Maschinensprache, Assembler oder einer höheren Sprache ausgebaut werden.

**Möglichkeit 4:** Aus grundsätzlichem Interesse oder aus wirtschaftlichen Erwägungen kann es nötig werden, einen $\mu$C, eine Steuerungs- oder Überwachungseinrichtung etc. aus optimal angepaßten Einzelkomponenten aufzubauen. D.h. man entscheidet sich für einen $\mu$P fester Wortlänge, einen *Einchip-$\mu$C* oder einen *Bitslice-Prozessor,* wählt die zusätzlich unbedingt notwendigen Bauteile aus und stellt damit sein ganz spezifisches System zusammen. Außerdem ist in diesem Fall gut zu überlegen, wie man das System zum „Laufen" kriegt und in welcher Weise Programme entwickelt und eingegeben werden können.

Ganz gleich für welche dieser oder ähnlicher Möglichkeiten man sich entscheidet oder zu welcher man gezwungen ist, es wird früher oder später nötig, weitere Informationen einzuholen oder ganz allgemein „den Horizont zu erweitern". Der Fachteil „Mikrocomputer" des Jahrbuches soll gerade dazu beitragen. In diesem Sinne wird z.B. der Beitrag über „Hilfsmittel zur Programmentwicklung und Testmethoden" vielseitige Horizonterweiterungen verschaffen können.

Eine immer größere Bedeutung erlangen die *Personal Computer.* Mit zwei Arbeiten wird darum dieses Thema aufbereitet. Und es spielt keine Rolle, ob man auf der „Hobby-Computer-Seite" tätig ist oder mehr professionell arbeitet, über „Schnittstellen und Busanordnungen sollte man einigermaßen Bescheid wissen. Der gleichnamige Übersichtsbeitrag kann in diesem Sinne nützlich sein.

In zwei Aufsätzen werden die grundlegenden Prinzipien der „Adressierung" und der „Interruptverarbeitung" diskutiert. Daran schließt sich ein Beitrag an, der die beiden wichtigen Verfahren zur „grafischen Darstellung von Programmabläufen" behandelt. Einen Schwerpunkt bildet der „Kompaktkursus BASIC". Mit erstaunlich geringem Aufwand wird hierdurch das Erlernen der wesentlichen BASIC-Grundlagen ermöglicht.

Es wird über die Ausbildung mit und an Mikrocomputern berichtet, ehe der Fachteil sich speziellen Problemen und $\mu$P-Systemen zuwendet. Hervorgehoben seien der praxisbezogene Aufsatz zum neuen „Interface Adapter 6522" und der Beitrag über „bipolare $\mu$P". Leser mit relativ umfangreichen Vorkenntnissen seien schließlich auf die Arbeiten mit einem Arithmetik-Prozessor hingewiesen.

Mikrocomputer sind auch nur Datenverarbeitungseinrichtungen. Wo aber liegen dann die Unterschiede zu anderen DV-Einrichtungen? Was bedeutet CPU, ALU, μP, μC, EDV-System, PDV? Erfahrene „Computing Fans" werden bei der Beantwortung dieser Fragen keine Schwierigkeiten haben. Für alle anderen Leser und für Freunde einer systematischen Zusammenstellung und Erklärung der wichtigsten Grundbegriffe wird dieser Beitrag dagegen von Nutzen sein. Aussagen über Anwendungsbereiche und darüber, wie μC eine festverdrahtete Logik ersetzen können, werden ebenfalls gemacht.

H. Schumny

# Mikrocomputer – Grundbegriffe und Abgrenzungen*)

## 1 EDV-System und Mikroprozessor

Der *Prozessor* ist die zentrale Steuer- und Verarbeitungseinheit einer Anlage für „automatische Datenverarbeitung". In dieser Einheit werden die einzelnen Programmbefehle interpretiert, daraus Steuerimpulse hergeleitet und, nach Maßgabe dieser Steueranweisungen, Daten verrechnet, also die gewünschten Ergebnisse ermittelt. Weil die Verarbeitung mit Hilfe elektronischer Schaltkreise vorgenommen wird, ist die Bezeichnung *Elektronische Datenverarbeitung* (EDV) allgemein eingeführt. In **Bild 1** ist für den Prozessor die Benennung „Steuer- und Rechenwerk" verwendet. Der englische Fachausdruck lautet *Central Processing Unit*, CPU. Die CPU bildet zusammen mit einem Speicher (*Memory*) die *Zentraleinheit* der Datenverarbeitungsmaschine. Vollständig ist eine EDV-Anlage (Computer, Rechner) aber erst, wenn eine ausreichende *Peripherie* mit Ein- und Ausgabegeräten angeschlossen ist.

Die Gesamtheit der in **Bild 1** schematisierten technischen Einrichtungen bildet die *Hardware* der EDV-Anlage. Fähig zum sinnvollen und selbsttätigen Arbeiten ist die Hardware aber erst dann, wenn eine konkrete Folge von Anweisungen vorliegt, die für die Maschine verständlich sind, die also entsprechend codiert sind. Solch eine Anweisungsfolge heißt *Programm*. Die Gesamtheit aller Programme, die teils vom Benutzer erstellt, teils vom Hersteller der Hardware mitgegeben sind, wird *Software* genannt. Erst die Verbindung von Hardware und „passender" Software ergibt ein funktionsfähiges *EDV-System*. Ohne

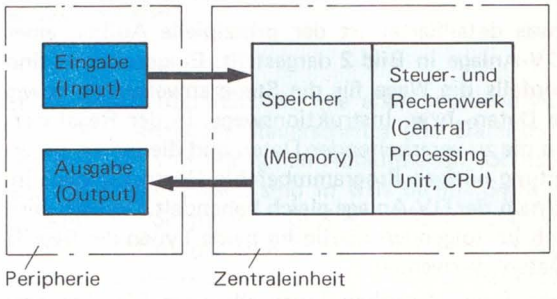



**Bild 1** Schematische Darstellung einer Datenverarbeitungsanlage

*) Vorabdruck aus *H. Schumny*, Mikrocomputer-Arbeitsbuch. Das Buch erscheint 1980 im Verlag Vieweg, Braunschweig/Wiesbaden.

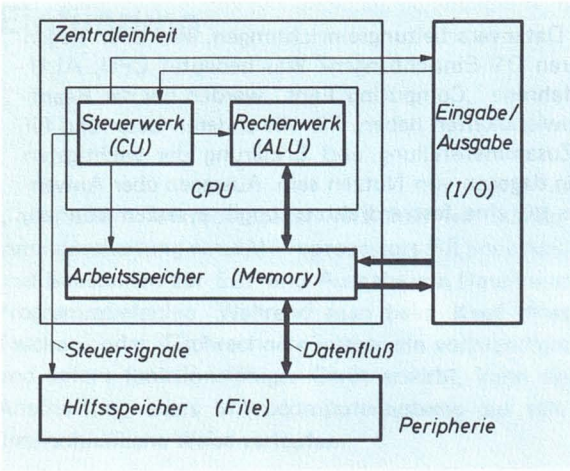



**Bild 2** Prinzipieller Aufbau eines EDV-Systems mit Steuersignalen und Datenfluß

Software bleibt die Hardware eine nutzlose Maschine. Es gilt somit die Formel:

> Ein EDV-System setzt sich zusammen aus Hardware und Software.

Etwas detaillierter ist der prinzipielle Aufbau einer EDV-Anlage in **Bild 2** dargestellt. Eingezeichnet sind ebenfalls die Wege für die Steueranweisungen sowie die Daten- bzw. Instruktionswege. In der Regel werden die zu verarbeitenden Daten und die für die Verarbeitung nötigen Programmbefehle (Instruktionen) innerhalb der DV-Anlage gleich behandelt. Deshalb wird auch im folgenden häufig für beide Typen der Begriff „Daten" verwendet.

Nach ihren Aufgaben innerhalb eines EDV-Systems werden die in **Bild 2** angegebenen Funktionseinheiten abgegrenzt:

> *Zentraleinheit* mit Steuerwerk (*Control Unit*, CU), Rechenwerk (*Arithmetic Logic Unit*, ALU) und Arbeitsspeicher (*Memory*). CU und ALU zusammen bilden die CPU (*Central Processing Unit*).
>
> *Peripherie* mit Ein-Ausgabe-Einheiten (*Input/Output Units*, I/O) und Hilfsspeicher (*File*).

Das *Steuerwerk* (auch: Leitwerk) ist die zentrale Überwachungs- und Steuerungseinrichtung der gesamten EDV-Anlage. Es gibt die Ein-Ausgabe von Daten und Programmbefehlen frei, sorgt für die Decodierung, also das Erkennen und Ausführen der Programmbefehle, und koordiniert sämtliche Geräte und Funktionseinheiten der EDV-Anlage.

Kontrolliert vom Steuerwerk nimmt der *Arbeitsspeicher* (auch: Hauptspeicher) von der Eingabeeinheit alle Daten und Programme auf und speichert sämtliche Zwischen- und Hauptergebnisse. Er ist also das zentrale Gedächtnis der Anlage. Arbeitsspeicher sind heute in der Regel als Halbleiterspeicher aufgebaut. Nur in älteren und größeren EDV-Anlagen findet man noch hin und wieder *Kernspeicher* (vgl. [1]).

Das *Rechenwerk* führt alle durch das Programm geforderten Verarbeitungen durch, wobei in der Hauptsache arithmetische und logische Verknüpfungen zu bewältigen sind, weshalb auch die Bezeichnung *Arithmetisch logische Einheit* gebraucht wird. Die zu verarbeitenden Daten werden auf Anweisungen des Steuerwerks hin aus dem Arbeitsspeicher in das Rechenwerk transferiert; die Ergebnisse gelangen zurück in den Arbeitsspeicher.

Ein zu großer Arbeitsspeicher wird in der Regel zu teuer und zu langsam sein; denn der Zugriff zu den einzelnen Speicherzellen ist um so langsamer, je größer der Speicher. Darum können zur Entlastung und Erweiterung des Arbeitsspeichers *Hilfsspeicher* dazugeschaltet werden. Nur die jeweils unmittelbar benötigten Daten und Programmbefehle müssen dann im Arbeitsspeicher direkt zur Verfügung stehen – sie müssen *speicherresident* sein. Wichtige Hilfsspeicher sind: Magnetplatten, Floppy Disks oder Disketten, Computerbänder, Magnetbandkassetten, Magnetblasenspeicher.

Die *Eingabe* von Programmen und Daten und die *Ausgabe* von Ergebnissen geschieht im wesentlichen auf eine der folgenden Weisen:

- *manuell* mit Hilfe von Tastaturen, Datensichtgeräten, Druckern oder mit Lochstreifen- bzw. Lochkartenlesern und Stanzern;
- *automatisch* (bzw. vom Steuerwerk veranlaßt) von und zu Meßstellen oder Datenträgern;
- per *Datenfernübertragung* von nahezu beliebig weit entfernten Außenstellen (*Terminals*).

> Wie **Bild 2** verdeutlicht, gelangen Programme und Daten in jedem Fall zuerst in den Arbeitsspeicher, nie direkt in einen Hilfsspeicher oder die CPU.

Wir wollen nun die einzelnen Steuerschritte beschreiben, die nötig sind, um Daten von der Eingabe-

einheit in den Arbeitsspeicher zu „laden". Für die in **Bild 2** dargestellten Funktionseinheiten sollen dabei folgende Abkürzungen benutzt werden:

Eingabeeinheit: IU (*Input Unit*)
Ausgabeeinheit: OU (*Output Unit*)
Steuerwerk: CU (*Control Unit*)
Rechenwerk: ALU (*Arithmetic Logic Unit*)
Arbeitsspeicher: MM (*Main Memory*)
Hilfsspeicher: FI (*File*)

Damit und mit den in **Bild 2** eingezeichneten Steuer- und Datenwegen kann die Aufgabe „Daten in Arbeitsspeicher laden" etwa folgendermaßen beschrieben werden:

1. IU teilt CU mit, daß Daten eingegeben werden sollen (Steuersignal);
2. CU fordert MM auf, Daten von IU zu übernehmen (Steuersignal);
3. CU teilt IU mit, daß nun Daten eingegeben werden können (Steuersignal);
4. IU versucht, Daten in MM zu laden (Datentransfer[1]));
5. Ist das Laden gelungen und abgeschlossen, meldet IU dies der CU (Steuersignal);
6. Ist das Laden nicht gelungen, laufen erneut die Schritte 1 bis 5 ab.

Es sei hervorgehoben, daß für den sehr einfachen Vorgang „Eingeben von Daten" bereits mindestens 5 Schritte auszuführen sind.

Der eben beschriebene Ladevorgang ist als typisch für alle normalen Abläufe in einer DV-Anlage anzusehen, weil er unter vollständiger Kontrolle des Steuerwerkes abläuft. Das hat zwei Konsequenzen: 1. Der Rechner ist während aller Ein-Ausgabevorgänge blockiert für andere Aufgaben, weil ja die CPU für diese Vorgänge benötigt wird; 2. Die „normalen" Ein-Ausgabevorgänge erfordern eine Reihe von CPU-Steuerschritten und damit viel mehr Zeit als die Ein-Ausgabe selbst. Wenn diese zusätzliche Zeit nicht zur Verfügung steht, weil die Daten zu schnell anfallen, oder wenn die CPU während der Ladevorgänge für andere Aufgaben genutzt werden soll, empfiehlt sich eine Spezialeinrichtung, die mit dem Fachausdruck „direkter Speicherzugriff" oder englisch als *Direct Memory Access* (DMA) bezeichnet wird. Es handelt sich um einen besonderen Datenkanal (DMA-Kanal), der es erlaubt, Daten ohne Inanspruchnahme der CPU in den Hauptspeicher zu bringen.

[1]) *Transfer* wird für „Übertragung" benutzt, um den Unterschied zum „körperlichen" Transport herauszustellen.

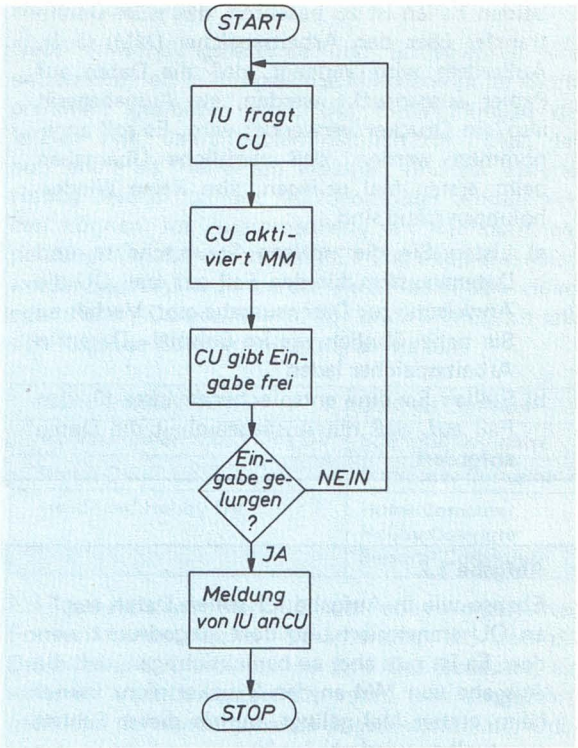



**Bild 3** Programmablaufplan für die Aufgabe „Daten in Arbeitsspeicher laden"

Es ist üblich und nützlich, die eben gegebene Auflistung der einzelnen Steuerschritte und Datentransfers in Form von grafischen Ablaufplänen darzustellen. Für diesen Zweck sind in DIN 66 001 Symbole genormt (s. *Schneider:* Grafische Darstellung von Programmabläufen). **Bild 3** zeigt den Programmablaufplan für unsere Aufgabe „Daten in Arbeitsspeicher laden".

Die ersten drei Schritte in **Bild 3** laufen nacheinander ab. Man nennt dies *lineare Anweisungsfolge* oder *sequentieller Ablauf*. In Schritt vier kommt es zu einer *Programmverzweigung*. An dieser Stelle wird eine *logische Entscheidung* gefordert. Wir werden hierauf später zurückkommen.

**Aufgabe 1.1**

Es sollen Daten aus dem Hilfsspeicher (FI) an die Ausgabeeinheit (OU) transferiert werden. Dies kann geschehen, indem das Steuerwerk (CU) entsprechend einer Programmanweisung die Aufforderung dazu gibt, oder wenn eine Anforderung der Ausgabeeinheit vorliegt. In

beiden Fällen ist zu beachten, daß jeder Datentransfer über den Arbeitsspeicher (MM) läuft. Außerdem wird verlangt, daß die Daten auf Papier ausgedruckt werden, als Ausgabegerät also ein Drucker verwendet wird. Es soll angenommen werden, daß sämtliche Übergaben beim ersten Mal gelingen, also keine Wiederholungen nötig sind.

a) Listen Sie die nötigen Steuerschritte und Datentransfers für den Fall auf, daß CU die Anweisung zur Datenausgabe gibt. Verfahren Sie dabei ähnlich wie im Beispiel „Daten in Arbeitsspeicher laden".
b) Stellen Sie eine entsprechende Liste für den Fall auf, daß die Ausgabeeinheit die Daten anfordert.

**Aufgabe 1.2**

Ebenso wie in Aufgabe 1.1 sollen Daten aus FI an OU transferiert und dort ausgedruckt werden. Es ist nun aber zu berücksichtigen, daß die Ausgabe von MM an den Drucker nicht immer beim ersten Mal gelingt, mithin dieser Schritt eventuell zu wiederholen ist.

Konstruieren Sie für diesen Fall den Programmablaufplan. Diese Aufgabe wird möglicherweise dadurch erleichtert, daß vorab eine wie in Aufgabe 1.1 geforderte Auflistung angefertigt wird.

**Mikroprozessor**

Wir haben in diesem ersten Abschnitt bislang überlegt, wie ein EDV-System prinzipiell aufgebaut ist, wie Steuersignale den gesamten Ablauf innerhalb des Systems lenken und welche Datenwege zur Verfügung stehen müssen. Die Darstellung der Zusammenhänge ist allerdings an dieser Stelle stark vereinfacht, sozusagen auf das Wesentliche beschränkt. Damit können wir aber leicht den Übergang zu unserem eigentlichen Thema „Mikroprozessor" angehen: Vereinfacht ausgedrückt, ist ein Mikroprozessor nichts anderes als die CPU eines Computers, der allerdings in seiner Leistungsfähigkeit, im direkt anschließbaren Speicherbereich und – nicht zuletzt – im Preis um einiges unter „normalen" Computern liegt. Bevor in späteren Kapiteln diese und weitere Merkmale von Mikroprozessoren verfeinert und ergänzt werden, soll hier folgende Begriffsbestimmung vorgenommen werden:

- *Mikrocomputer* nennt man die Zusammenschaltung einer CPU mit speziellen, zumeist eingeschränkten Fähigkeiten mit Speichern und Ein-Ausgabegeräten;
- *Mikroprozessor* ist die CPU eines Mikrocomputers;
- *Mikrocomputer-System* ist die Verbindung eines Mikrocomputers (Hardware) mit Programmen und Programmierungshilfen (Software).

## 2 Anwendungsbereiche und Rechnerklassen

Um besser verstehen zu können, wo die Stärken und Grenzen von Mikroprozessoren liegen, sind allgemeine Abgrenzungen von Anwendungsbereichen und Rechnerklassen nützlich. Wir beginnen diese Abgrenzungen folgendermaßen:

*Hauptbereiche der maschinellen oder automatischen Datenverarbeitung* (ADV) sind
- kommerzielle Datenverarbeitung
- technisch-wissenschaftliche Datenverarbeitung
- Prozeßdatenverarbeitung (PDV)
- Analogdatenverarbeitung

Neben den „klassischen" Hauptbereichen der ADV (kommerziell und technisch-wissenschaftlich) sind in unsere Betrachtung die „modernen Zweige" PDV und Analogdatenverarbeitung aufgenommen. Dazu im folgenden mehr.

Typisch für *kommerzielle DV* ist einerseits eine Vielzahl von Daten, die Speicher sehr großer Kapazität erfordern (sogenannte Massenspeicher bzw. Datenbanken). Andererseits ist in der Regel nur ein geringer Aufwand an Verrechnung nötig (relativ einfache und kurzzeitige Rechenoperationen). Als Beispiel stelle man sich die Kontenführung eines großen Bankinstitutes vor mit vielleicht 100 000 Kunden. Bei reger Benutzung des Banken-Service (Überweisungen, Abhebungen, Daueraufträge, Schecks etc.) ergibt sich eine enorme Zahl von Daten (neuer Kontostand, Gebühren, Soll, Haben, Zinsen, Mahnungen etc.). Die Verrechnung dieser Daten und die Ermittlung des aktuellen Standes jedoch besteht aus einfachsten Operationen wie Addition und Multiplikation.

Gewissermaßen konträr zur kommerziellen DV, also in umgekehrtem Verhältnis erweist sich *technisch-wissenschaftliche DV*. Die in diesem Hauptbereich be-

handelten Probleme sind fast immer charakterisiert durch komplexe, umfangreiche und somit langwierige Verrechnungen weniger Ein- und Ausgabedaten. Natürlich handelt es sich bei diesen Unterscheidungen um Grenzfälle. In der Praxis wird sich oft eine mehr oder weniger starke Vermischung einstellen.

Eine besondere, aber immer wichtiger werdende Form in sowohl technisch-wissenschaftlichen als auch kommerziellen Bereichen wird als *Dialog-Datenverarbeitung* (DDV) bezeichnet. In „normalen" Anwendungsfällen wird zur Lösung eines bestimmten Problems ein Programm geschrieben, das vollständig ablaufen muß, um ein Ergebnis zu liefern. Erst danach ist es möglich, das Programm gegebenenfalls zu ändern, um ein „besseres" Ergebnis zu erzielen, was aber erneut einen vollständigen Programmablauf erfordert, usw. Bei DDV dagegen kann die Lösung eines Problems in einzelnen Schritten erfolgen, weil nicht erst das ganze Programm ablaufen muß, sondern der Rechner auf jede Anfrage direkt eine Antwort erteilt (Dialog Mensch/Maschine). Zu diesem Zweck muß natürlich der Hersteller den Rechner speziell ausgelegt haben, ihm also ein darauf zugeschnittenes *Betriebssystem* mitgegeben haben (vgl. hierzu [1] und [2]).

Von *Prozeßdatenverarbeitung* (PDV) spricht man dann, wenn EDV-Anlagen in technischen oder wissenschaftlichen Prozessen für z.B. folgende Aufgaben eingesetzt werden:

- Meßdatenerfassung und -auswertung;
- Überwachung und Steuerung technischer Produktionsprozesse (Prozeßführung);
- Prozeßregelung.

Beispiele hierfür sind heute in allen Forschungs- und Entwicklungslaboratorien sowie in den verschiedenartigsten Produktionsanlagen zu finden. Ein typisches Merkmal für PDV ist, daß die zumeist analogen Meßdaten (Prozeßdaten) vor der Verarbeitung „digitalisiert" werden müssen. Dazu werden dem Rechner Analog-Digital-Wandler (ADC, *Analog Digital Converter*) vorgeschaltet. Die Ergebnisse der digitalen Verarbeitung werden entweder direkt weiterverwendet (*Closed Loop,* d.h. „geschlossene Regelschleife"), oder sie werden über einen DAC (*Digital Analog Converter*) wieder ausgegeben. Häufig anzutreffen ist der Fall, daß Prozeßdaten in *Echtzeit* zu verarbeiten sind, daß also die Verarbeitungsergebnisse vor dem nächsten Prozeßschritt zur Verfügung stehen müssen. Schließlich sind oft sehr große Datenmengen in manchmal schneller Folge (hohe Datenraten) zu verkraften.

**Hobby-Computing**

Neue Anwendungsbereiche und Schlagwörter sind entstanden, seit leistungsfähige Mikrocomputer enorm preiswert geworden sind. Die Entwicklungen der letzten zwei Jahre hatten nämlich zur Folge, daß nun auch als Helfer im Haushalt und für vielerlei Hobby-Beschäftigungen Mikrocomputer genutzt werden können. Im Zusammenhang mit Rechnern, wie sie beispielsweise durch den PET 2001 charakterisiert werden, spricht man darum heute auch von Heim- oder Hobby-Computern (s. [3] und [4]). Damit läßt sich folgende Zusammenfassung aufstellen:

| | |
|---|---|
| Kommerzielle DV | Business Computer |
| technisch-wissenschaftliche DV | Scientific Computer |
| Dialog-DV | Interactive Computer |
| Heim- und Hobby-DV | Home Computer<br>Hobby Computer<br>Personal Computer |

Ein gemeinsames Merkmal der eben vorgestellten Formen automatischer Verarbeitung ist, daß die Daten *seriell* verarbeitet werden. Das bedeutet, ein weiterer Verarbeitungsschritt wird erst ausgeführt, nachdem der vorhergehende abgeschlossen ist. Oder anders beschrieben: die zur Lösung eines Teilproblems nötigen Verarbeitungsschritte werden in einem einzigen „Rechenwerk" nacheinander (also seriell) ausgeführt. Die alternative Möglichkeit ist, ebensoviele „Rechenwerke" zu installieren, wie Verarbeitungsschritte zur jeweiligen Problemlösung erwartet werden. Dieses aufwendige *parallele Verfahren* wird in Digitalrechnern nicht angewendet, ist jedoch typisch für *Analogrechner.*

Im Gegensatz zu Digitalrechnern, die *diskrete Werte* (seriell) verarbeiten, arbeiten *Analogrechner* mit im wesentlichen stetigen physikalischen Größen (vgl. z.B. [5]). Die Lösung eines Problems wird hierbei nicht ziffernweise (numerisch) herbeigeführt, sondern es wird sozusagen ein physikalisches (analoges) Modell des Problems aufgebaut und dann die Lösung der Aufgabe durch Experimente mit dem analogen System gewonnen und durch Messungen der physikalischen Größen des Systems registriert. Das analoge Modell besteht darin, daß die mathematisch formulierte Aufgabe in *Rechenelemente* umgesetzt wird – Rechenelemente für Addition, Subtraktion, Multiplikation, Integration, Differentiation usw. der analogen Signalverläufe. Es müssen soviele Rechenelemente vorhanden sein, wie es die Aufgabe bzw. die gewünschte Genauigkeit fordert. Die Verarbeitung (Problemlösung) erfolgt dann *parallel*, also in einem Schritt mit allen

einbezogenen Rechenelementen. Analogrechner sind darum „schneller" als Digitalrechner. Wo es aber auf hohe Genauigkeit ankommt, wird der Digitalrechner eingesetzt werden. Um die Vorteile beider Rechentypen nutzen zu können, werden *Hybridrechner* aufgebaut. Sie enthalten außer den beiden Teilsystemen Analogrechner und Digitalrechner eine *Kopplungselektronik* zur „Vermittlung" zwischen ihnen.

Die „konventionelle Technik" zum Entwickeln und Aufbauen digitaler Schaltkreise besteht darin, daß das insgesamt gewünschte Verhalten durch Zusammenschaltung diverser integrierter Bausteine (IC, *Integrated Circuit*) erzielt wird. Das bedeutet, es werden aus einer „IC-Logikfamilie" diejenigen Bausteine herausgesucht, die zur Lösung der jeweiligen Aufgabe nötig erscheinen. Das gewünschte Verhalten (die „Intelligenz" des Schaltkreises) ergibt sich aus der richtigen Verknüpfung der Logik-Bausteine. **Bild 4** zeigt schematisch den Signalfluß solch eines konventionellen Systems. Solche „festverdrahteten Logiksysteme" haben offensichtlich den Nachteil, daß bei einer Veränderung der Aufgabenstellung auch neu „verdrahtet" werden muß. Bei Großserien bedeutet dies eine Änderung der bedruckten Platten.

**Bild 4**

In digitalen Schaltkreisen mit Mikroprozessoren ist die „festverdrahtete Logik", die sinnvolle Zusammenschaltung einzelner Logik-Bausteine also, ersetzt durch einen Mikroprozessor. Das hat folgende Konsequenzen:

- digitale Schaltkreise werden im wesentlichen mit nur noch einem Baustein aufgebaut – dem $\mu$P;
- die „Intelligenz" des Schaltkreises, das gewünschte Verhalten also, wird durch ein Programm erzeugt;
- die Anpassung an eine veränderte Aufgabenstellung wird durch Änderungen im Programm erzielt.

**Bild 5** soll im Vergleich mit **Bild 4** den Unterschied zur konventionellen Technik verdeutlichen.

> In digitalen Schaltkreisen mit Mikroprozessoren ist die konventionelle, festverdrahtete Logik (*Hardware*) ersetzt durch eine Folge von Anweisungen, die der $\mu$P ausführt (*Software*). Bei Veränderungen der Anforderungen wird nur die Software geändert; Eingriffe in die Hardware sind nicht mehr nötig.

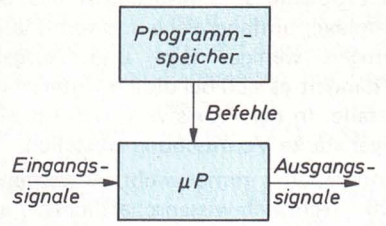



**Bild 5**

Daraus folgt nun aber als oft schwerwiegende Konsequenz:

> Die entscheidende Arbeit beim Aufbau von Schaltkreisen mit $\mu$P muß in der *Programmierung* gesehen werden. Der „klassische" Digitaltechniker wird mithin – zumindest partiell – zum Programmierer!

> **Aufgabe 1.3**
> Wir haben festgestellt, daß in konventionellen digitalen Schaltkreisen die „festverdrahtete Logik" durch einen Mikroprozessor ersetzt werden kann. Überlegen oder diskutieren Sie dazu folgendes:
> a) Wann halten Sie das Ersetzen einer festen Verdrahtung durch einen $\mu$P für sinnvoll (vergleichbare Hardware-Kosten angenommen), bei einer einzelnen Schaltung, bei einer kleinen Serie oder bei einer Großserie?
> b) Durch welche Arbeitsgänge werden die hauptsächlichen Kosten verursacht?
> c) Zählen Sie auf, welche Fälle aus Ihrer Umgebung (oder ganz allgemein) für eine $\mu$P-Verwendung geeignet erscheinen. Denkbar ist dabei natürlich, daß durch den $\mu$P nur jeweils Funktionsteile ersetzt werden.

**Literatur**

[1] *Schumny, H.:* Digitale Datenverarbeitung für das technische Studium. Viewegs Fachbücher der Technik, Braunschweig 1975.

[2] *Schumny, H.:* Signalübertragung, Lehrbuch der Nachrichtentechnik und Datenfernverarbeitung. Viewegs Fachbücher der Technik, Braunschweig 1978.

[3] *Schlossberg, E., Brockman, J.* und *Horton, L.:* The Home Computer Handbook. Bantam Books Inc., New York 1978.

[4] *Zaks, R.:* An Introduction to Personal and Business Computing. Sybex Inc., Paris 1978.

[5] *Hannauer, G.:* Grundlagen des hybriden Analogrechnens. R. Oldenbourg Verlag, München 1973.

"Personal Computer" könnten in den nächsten Jahren eine ähnliche Entwicklung erfahren, wie sie in den vergangenen Jahren bei den Taschenrechnern zu beobachten war. Dieser Beitrag erhärtet diese Vermutung mit einer Reihe von Zahlen und Entwicklungstendenzen. Aufgezeigt wird außerdem folgendes: Obwohl Personal Computer die gleiche Grundstruktur wie alle anderen informationsverarbeitenden Systeme aufweisen, liegt eine Besonderheit darin, daß sie im wesentlichen interaktiv, also im Dialog genutzt werden, ohne von ihrem Anwender irgendwelche Systemkenntnisse zu verlangen. Einsatzmöglichkeiten werden an den Bereichen Planen, Lernen und Gestalten diskutiert.

H. v. Falser

# Planen – Lernen – Gestalten

## Das Programm des Personal Computers

Wer die kurze Entwicklungsgeschichte elektronischer Rechner während der letzten 30 Jahre betrachtet, dem wird schnell klar, daß die entscheidenden Durchbrüche in der Leistung und im Verständnis *Digitaler Informationsverarbeitender Systeme* nicht die Folge geplanter **Evolution** sondern vielmehr Ergebnis der wesentlichen Stationen auf dem Gebiete der Halbleiterelektronik waren.

Als anfangs der fünfziger Jahre die ersten großen *Röhrenrechner,* bis dahin Adoptivkinder geheimer militärischer Forschungsstäbe, auch für komplizierte technisch-wissenschaftliche Berechnungen außerhalb der Kriegsforschung eingesetzt wurden, waren sich die meisten Experten darüber einig, daß mit 10 bis 20 solcher Großanlagen der Rechenbedarf der Vereinigten Staaten auf absehbare Zeit abgedeckt werden könnte.

Sehr viel anders wäre es wahrscheinlich auch nicht gekommen, hätte nicht bereits 1943 in den Laboratorien der Bell-Telephon-Corporation in Murray Hills (USA) eine Gruppe von Physikern, Chemikern und Metallurgen unter *William B. Shockley* die grundsätzliche Frage nach den Zukunftschancen der Elektronenröhre gestellt und dabei die Erkenntnis gewonnen, daß dieser wohl keine wesentlichen technologischen Fortschritte mehr beschieden sein würden. Shockley's Team hatte sich deshalb darauf konzentriert, den bereits 1874 von *Ferdinand Braun* in Leipzig entdeckten sogenannten **Halbleitereffekt** zur Entwicklung von neuartigen elektronischen Schaltungen zu verwenden.

1948 entstand nach langen Jahren experimenteller Arbeit der *Transistor* – bereits 1952 begann die kommerzielle Karriere der Halbleiterelektronik. Kurz darauf wurde die erste transistorisierte elektronische Rechenanlage, die TX-O gebaut, und der Elektronenrechner machte sich auf, die Forschungslabors, Verwaltungszentren und Produktionsstätten der Industrie zu erobern.

In den darauffolgenden 20 Jahren bis zum Beginn der 70er Jahre konnten, immer auf der Basis der Transistorenelektronik, die systematischen Grundlagen der industriellen und kommerziellen Datenverarbeitung geschaffen werden, die aus unserem täg-

**Evolution:** Fortschreitende Entwicklung, also geplante Verbesserung und Vervollkommnung bzw. die bestimmten Gesetzen gehorchende Veränderung.

**Halbleitereffekt:** Die Tatsache, daß an Grenzflächen zwischen Metallen und Halbleitern bzw. zwischen *n*- und *p*-leitenden Halbleitern je nach Vorzeichen der angelegten Spannung ein elektrischer Strom fließen kann oder gesperrt wird.

lichen Leben als Organisation und Privatmensch nicht mehr wegzudenken sind.

Seit ungefähr 3 Jahren wissen wir, daß wir an der Schwelle der 80er Jahre wiederum den Beginn einer neuen Aera miterleben werden, in der nicht nur unsere technische Umwelt, sondern auch (heute nur teilweise absehbare) weite Bereiche unserer privaten Informationsverarbeitung von der Weiterentwicklung jener kleinen elektronischen Gehirnbausteine geprägt sein werden, die 1970 in Kalifornien unter dem Namen *Mikroprozessor* das Licht der Welt erblickt haben.

Das Durchschauen ihres komplizierten wirtschaftlich-technischen Eigenlebens geht nur wenige von uns an. Für jeden von uns wichtig dürfte dagegen langfristig das Verstehen des Konzeptes und der Möglichkeiten der neuen elektronischen Geräte werden, die als "Personal Computer" (**Individuelle Dialogrechner**) um die verschiedenen Mikroprozessorarchitekturen im Entstehen sind und mit denen der elektronische Rechner sich anschickt, heute den Schritt vom industriellen zum individuellen Anwender zu machen.

Die Chance, aus dem Elektronenrechner einen freundlichen, verläßlichen Partner und Berater beim Planen, Lernen und Gestalten zu machen, werden wir jedoch nur nützen können, wenn es uns gelingt, die Kommunikation zwischen Mensch und Maschine im Laufe der nächsten Jahre in entsprechend kreativer und effizienter Weise entscheidend zu verbessern.

Wir wollen uns zunächst kurz dem technischen Konzept des individuellen Dialogrechners zuwenden, um seine Fähigkeiten kennenzulernen, und den Rest dieser Einführung dem Problem der individuellen **Anwendungssoftware** widmen.

Alle digitalen informationsverarbeitenden Systeme haben eine wie in **Bild 1** dargestellte gemeinsame Grundstruktur.

Rechner, Speicher und Ein-Ausgabeschnittstellen lassen sich allgemein auf Systeme elektronisch gesteuerter Schalter zurückführen, die durch Baugruppen, bestehend aus Transistoren und dazugehöriger Schaltelektronik, realisiert werden. Im Bereich dieser Baugruppen – *Active Element Groups* oder AEG – hat sich im wesentlichen die Geschichte elektronischer Rechenanlagen abgespielt. Die Kosten pro AEG für sogenannte *Transistor Transistor Logik* (TTL) konnten in den letzten 10 Jahren um den Faktor 60 reduziert werden, die der gesamten sogenannten diskreten Baugruppen, d.h. auf Leiterplatten verschalteter AEG's in Form von Prozessoren und Speichern lediglich um den Faktor 15 (**Bild 2**).

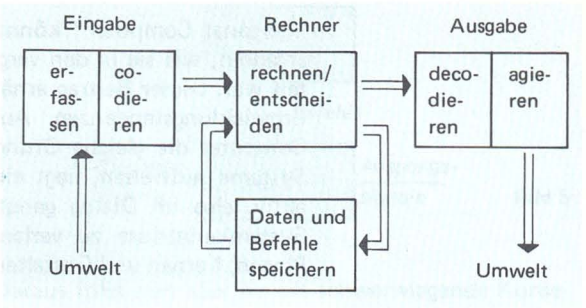



**Bild 1** Grundstruktur digitaler informationsverarbeitender Systeme

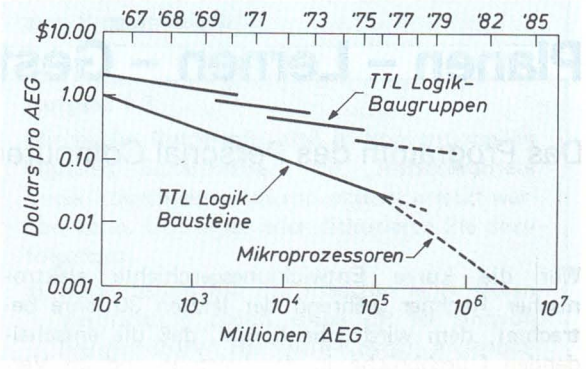



**Bild 2** Entwicklung der Kosten pro Active Element Group (AEG) für TTL-Baugruppen und Mikroprozessoren

Die Elektronikindustrie, bis ca. 1971 vorwiegend Hersteller eben dieser diskreten logischen Baugruppen immer höherer Komplexität, setzt seit der Einführung der sogenannten **Large Scale Integration** (Mikrominiaturisierung) in eben diesem Zwischenbereich an, in dem sie durch neue Produktionstechniken eine immer

**Individueller Dialogrechner:** Mit dieser Bezeichnung wird ausgedrückt, daß *Personal Computer* am vorteilhaftesten *interaktiv,* also im *Dialog zwischen Mensch und Maschine* genutzt werden.

**Anwendungssoftware:** Programme, mit denen der Anwender aus dem Universalrechner ein Hilfsmittel zur Lösung seiner individuellen Aufgaben machen kann.

**Large Scale Integration:** LSI. Beschreibung hochentwickelter Halbleitertechnologien, die mehr als 100 Schaltfunktionen auf einem Halbleiterplättchen (*Chip*) realisieren.

höhere Anzahl von AEG's auf einem einzigen Siliziumscheibchen unterbringt und damit *Integrierte Schaltkreise* (IC) mit immer höherer Dichte von Funktionen herstellt. Dies wirkt sich insbesondere in zwei Bereichen aus, nämlich in dem der Mikroprozessoren, wo arithmetische und logische Rechenwerke zusammen mit den erforderlichen schnellen Zwischenspeichern als freiprogrammierbare Bauelemente angeboten werden, und im Bereich der elektronischen Speicher selber, wo die Ferritkernspeicher der 50er, 60er und frühen 70er Jahre jetzt zügig durch vollelektronische Medien abgelöst werden.

Die Kosten für sogenannte dynamische Schreib-Lesespeicher oder RAM's sind von 0,3 Cent pro Bit im Jahre 1972 heute bereits auf 0,1 Cent pro Bit gesunken (**Bild 3**), und man erwartet einen weiteren Faktor 10 an Kostenersparnis im Laufe der nächsten 10 Jahre. Unter diesen Annahmen kann man für 1985 die Kosten für 1 Million Byte auf weniger als 2000 $ projizieren.

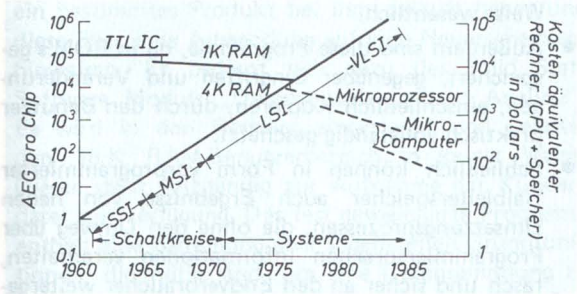



**Bild 3** Kostenentwicklung bei Speichern (Preis/Leistungsverhältnis)

Sieht man den Mikroprozessor im wesentlichen als einen freiprogrammierbaren, logischen Baustein an, so ergibt sich daraus eine Kette von Konsequenzen:

- Derselbe Satz von Bauelementen kann für eine große Zahl praktisch maßgeschneiderter Anwendungen eingesetzt werden.
- Daraus ergeben sich eine sehr hohe Stückzahl bei der Produktion und entsprechend günstige Einzelkosten.
- Der Systementwickler kann somit schnell und kostengünstig individuelle Geräte konzipieren, da sich seine Arbeit auf Programmierung und Anpassung von Schnittstellen konzentrieren kann.
- Durch die hohe Verfügbarkeit von Hardware wird ein bisher unerreichter Grad von Benutzerfreundlichkeit möglich, vorausgesetzt, es steht das entsprechende Knowhow zur Verfügung.

  In diesem Umfeld entsteht zur Zeit der rasch wachsende Markt des *Personal Computers.*

In seiner heutigen Form eignet sich der Personal Computer für Aufgaben, die mit **Dateien** von nicht mehr als 1–2 Millionen Zeichen im **Direktzugriff** arbeiten. Darüber entsteht ein Mißverhältnis zwischen der Leistungsfähigkeit der Massenspeicher und der Zentraleinheit. Mit einem solchen System, dem ein Arbeitsspeicher von typisch 16 Kbyte zur Verfügung steht, lassen sich trotzdem viele wichtige Anwendungen aus den Bereichen *Planen, Lernen* und *Gestalten* in das unmittelbar erreichbare private oder berufliche Umfeld individueller Anwender bringen.

Der Schlüssel zum Erfolg der neuen Rechnergeneration liegt aber nach wie vor bei dem Problem der Kommunikation zwischen Rechner und Anwender, also bei der Erstellung individueller Anwendungssoftware.

Im Verhältnis zu den Entwicklungen des Preis/Leistungsverhältnisses der Hardware nehmen sich die Fortschritte seit 1950 bei der Programmierung elektronischer Rechner äußerst bescheiden aus (**Bild 4**).

Die Anwendungsprogramme sind zwar etwas kürzer und kompakter und auch für den Laien aussagefähiger geworden, aber die Schranke zu einer echten Massenverbreitung von elektronischer Rechenleistung erscheint heute fast so hoch wie ehemals. Zumal es äußerst unwahrscheinlich scheint, daß wir in naher Zukunft allen unseren Zeitgenossen das Programmieren beibringen oder aber die endgültig natürliche Programmiersprache erfinden werden.

Ein anderer, sehr innovativer aber ebenso aussichtsreicher Weg besteht darin, Software in Form von

**Datei:** Organisatorische Einheit einer Menge von Daten, die nach Verabredung geordnet und gekennzeichnet sind.

**Direktzugriff:** Zugriffsart, die dem Benutzer praktisch sofort gewünschte Daten von einem Datenträger (z.B. Magnetplatte) zur Verfügung stellt. Die Magnetbandkassette erlaubt keinen Direktzugriff.

| MASCHINEN SPRACHE | ASSEMBLER SPRACHE (MIX) | | |
|---|---|---|---|
| + 001100249 | START | ENT1 | I |
| + 009900250 | | ENT2 | N |
| + 100500039 | | JMP | 2F |
| + 120000539 | 5H | JE | SUCCESS |
| + 000130249 | | ENT1 | 1.3 |
| + 000010248 | 2H | ENTA | 0.1 |
| + 000020048 | | INCA | 0.2 |
| + 000100706 | | SRB | 1 |
| + 200000524 | | STA | TEMP |
| + 200000557 | | CMP1 | TEMP |
| + 120200639 | | JG | FAILURE |
| + 200000511 | | LD3 | TEMP |
| + 200100508 | | LDA | K |
| + 200230556 | | CMPA | KEY 3 |
| + 100300739 | | JGE | 5B |
| + 000130250 | | ENT2 | 1.3 |
| + 100500039 | | JMP | 2B |

FORTRAN

```
      FIRST = 1
      LAST  = N
      FOUND = TRUE
10    THIS = (FIRST + LAST)/2
      IF (A (THIS) = X) 20, 40, 30
20    FIRST = THIS + 1
      GOTO 35
30    LAST = THIS + 1
35    IF (FIRST.LE.LAST) GO TO 10
      FOUND = FALSE
40    CONTINUE
```

PASCAL

```
FIRST = 1  LAST = N  FOUND = FALSE
PEPEAT
      THIS = (FIRST + LAST) DIV 2;
      IF A (THIS) = X  THEN  FOUND = TRUE
      ELSE
         IF A (THIS) < X THEN FIRST = THIS + 1
         ELSE LAST = THIS + 1
UNTIL FOUND OR (FIRST = LAST)
```

**Bild 4** Gegenüberstellung des Programmieraufwands

Hardware anzubieten. Halbleiterlesespeicher (*Read Only Memories,* ROM) beginnen gegenüber konventionellen Datenträgern (Lochkarten, Magnetbändern, Plattenstapeln) immer mehr Fuß zu fassen. Von der Kostenseite sind sie bereits bei programmierbaren Taschenrechnern mit Anwendungen um 3000 Programmschritte seit einigen Jahren sehr erfolgreich.

Für den Personal Computer sind Anwendungsprogramme bis 30 Kbyte ökonomisch vertretbar und seit kurzem verfügbar. Anwendungsprogramme bis 500 Kbyte werden bereits im Labor getestet. ROM oder *Solid State Software* (ein Markenname von Texas Instruments) hat zwar heute noch einen Produktionszyklus von ca. 3 Monaten, bringt jedoch eine Reihe unmittelbar einleuchtender Vorteile:

- Ihre Vervielfältigung unterliegt den Gesetzen der Halbleiterproduktion und kann entsprechende Kostenvorteile unmittelbar weitergeben.
- Periphere Programmspeicher, ein wesentlicher Kosten- und Störfaktor bei heutigen Systemen, können schrittweise durch *Steckmodule* abgelöst werden.
- Techniken der *strukturierten Programmierung* werden von der Hardware vorgegeben, während sie im Bereich konventioneller Programmierung oft nur schwer durchzusetzen sind.
- Die Anwendungsprogramme stehen nach dem Einschieben der Steckmodule in den Rechner unmittelbar, d.h. ohne jede Ladezeit zur Verfügung und erleichtern den Dialog mit dem Benutzer auf diese Weise wesentlich.
- Außerdem sind diese Programme, da in ROM's gespeichert, gegenüber Eingriffen und Veränderungen, einschließlich Kopieren, durch den Benutzer praktisch vollständig geschützt.
- Schließlich können in Form vorprogrammierter Halbleiterspeicher auch Ergebnisse von neuen Umsetzungsprozessen, die ohne den Umweg über Programmiersprachen Informationen verarbeiten, rasch und sicher an den Endverbraucher weitergegeben werden. Ein neues aufsehenerregendes Beispiel ist die *Synthetisierung des gesprochenen Wortes* und anderer akustischer Informationen, deren beginnender Einsatz in Rechnerprodukten bereits heute eine neue wesentliche Aera der Kommunikation zwischen Mensch und Maschine eingeleitet hat.

*Solid State Software* ist verhältnismäßig billig, absolut zuverlässig, aber in ihrer angebotenen Form unveränderlich. Daraus darf jedoch keineswegs der Schluß gezogen werden, daß dem individuellen Anwender lediglich die Verwendung von Standardprogrammen zur Verfügung steht. Dies wollen wir im folgenden an 3 Beispielen aus den wesentlichsten Einsatzbereichen des Personalcomputers – Planen-Lernen-Gestalten – demonstrieren.

Als Demonstrationsobjekt soll uns dazu die Standardausführung des neuen Heimcomputer-Systems

TI 99/4 von Texas Instruments dienen. Dieser enthält einen 16 Byte Mikroprozessor aus der 9900-Familie, 16 K Anwender – RAM-Speicher, eine normale, Schreibmaschinentastatur und eine Vorrichtung zur Aufnahme von Solid State Software-Steuermodulen von 6 bis 30 Kbyte Kapazität. Zur Ein- und Ausgabe dient ein Farbfernsehgerät, das über einen Videokanal angesteuert wird und über Programm sowohl Zahlen, Zeichen als auch graphische Symbole und 16 verschiedene Farben wiedergeben kann. Der Rechner enthält außerdem standardmäßig ein Modul zur Wiedergabe von Tönen über 3 Oktaven, Geräuschen sowie, als Option, synthetisch gespeicherte Sprache in Form von Einschüben mit je 200 bis 300 Worten. Diese können über einen normalen Niederfrequenzverstärker mit angeschlossenem Lautsprecher wiedergegeben werden.

## Planen:

Ein Handelsunternehmer will sich informieren, welche seiner Kunden im Laufe der letzten 3 Jahre ein bestimmtes Produkt bei ihm gekauft haben und diese über eine Aussendung auf eine Neuerscheinung hinweisen. Er bedient sich dazu des Solid State Software Moduls „Datenverwaltung und Analyse". Es wird in den Rechner eingesteckt und belastet den 16 K RAM-Benutzerbereich in keiner Weise. Dieser steht vollständig zur Aufnahme der Kundendatei zur Verfügung. Das fest gespeicherte Programm enthält eine Sammlung aller sogenannten Grundfunktionen, die nötig sind, um eine Datensammlung zu erzeugen, um entsprechende Sätze darin aufzurufen, zu verändern und wieder abzuspeichern und um Ein- und Ausgabeformate durch den Anwender definieren zu lassen. Darüberhinaus enthält es eine große Anzahl von Analysefunktionen, die dazu dienen, Daten nach bestimmten Kriterien zu suchen und mathematische Operationen mit ihnen durchzuführen. Der Ablauf des Programmes ist völlig dialoggesteuert. Dies bedeutet, daß der Benutzer über den Bildschirm lediglich eine Folge von Nachrichten, sogenannten *Displays,* erhält, die ihn auffordern, entweder Entscheidungen zu treffen und damit die gewünschte Funktion anzuwählen (**Bild 5**), oder Daten und Parameter einzugeben (**Bild 6**).

Der Anwender hat nun die Möglichkeit, aus den bereitgestellten Funktionen eine völlig auf seine individuellen Bedürfnisse zugeschnittene Transaktion zu entwickeln. Dies geschieht über einen speziellen Speicherbereich, der sogenannte **File Header,** der

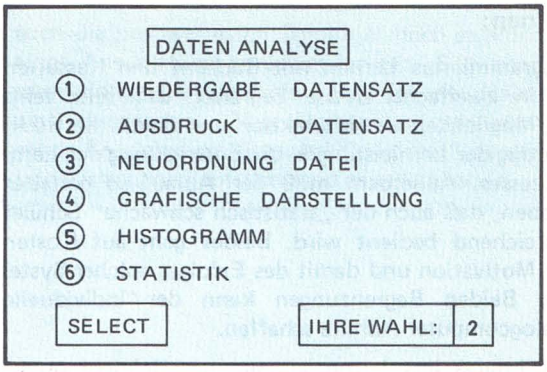



**Bild 5** Display-Beispiel: Aufforderung zur Auswahl einer Funktion

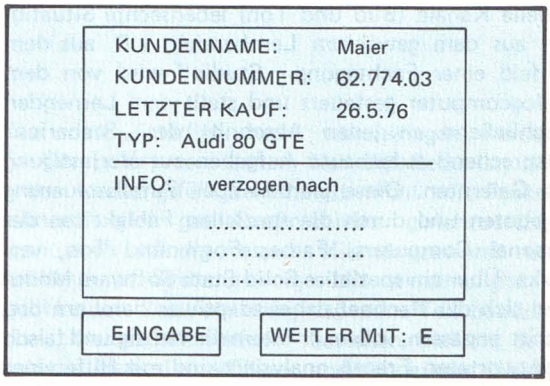



**Bild 6** Display-Beispiel: Aufforderung zur Daten- und Parametereingabe

der physikalischen Datei voran gespeichert wird und in dem eine Reihe von Benutzerparametern abgelegt sind. Dies sind einmal Informationen über die *Formate* der Ein- und Ausgabe an Displays, die der Benutzer wie bei der Erstellung eines Formulars auf einer Schreibmaschine über die Tastatur einfach auf den Bildschirm niederschreibt. Außerdem können hier alle Entscheidungen des Benutzers während eines Ablaufes in einem **Answer File** festgehalten werden. Damit läßt sich ein bestimmter Programmablauf beim ersten Mal individualisieren und später beliebig oft automatisch nachvollziehen.

**File Header:** Kopf (*Header*) einer Datei (*File*) mit diversen Angaben über die Datei.

**Answer File:** Datei der gewählten Verzweigungen, die während eines Programmablaufs erstellt oder ergänzt wird und (so aktualisiert) für weitere Verwendungen zur Verfügung steht.

## Lernen:

Programmiertes Lernen mit Büchern und Kassetten ist in zweifacher Weise begrenzt. Einerseits fehlt die Möglichkeit der Interaktion und damit die Rückführung der Lernleistung in die Entwicklung des Lernprozesses. Außerdem muß der Ablauf so gestaltet werden, daß auch der „statistisch schwache" Schüler ausreichend bedient wird. Beides geht auf Kosten der Motivation und damit des Erfolges solcher Systeme. Beiden Begrenzungen kann der individuelle Dialogcomputer Abhilfe schaffen.

Ein interaktives Lernprogramm am Heimcomputer, beispielsweise ein Englischkursus, besteht sinnvollerweise aus 2 Untersystemen: dem „Scenario" und dem „Studio". Das „Scenario" bietet über normale audiovisuelle Kanäle (Bild und Ton) lebensechte Situationen aus dem gewählten Lernbereich, z.B. aus dem Umfeld einer Fachtagung. „Studio" wird von dem Dialogcomputer gesteuert und stellt dem Lernenden anschließend an jeden Abschnitt des „Scenarios" entsprechend aufgebaute Aufgaben zur Verfestigung des Gelernten. Diese werden über Zufallssteuerung angeboten und durch die speziellen Fähigkeiten des Personal Computers, Farbe, Form und Ton, verstärkt. Über ein spezielles Solid State Software Modul wird sich der Rechner dabei adaptiv an den Lernfortschritt anpassen, indem er intern die richtig und falsch beantworteten Fragen analysiert und mit Hilfe eines Eliminationsverfahrens den weiteren Verlauf der Studiositzung so steuert, daß diese sich automatisch auf diejenigen Teile des Lernstoffes konzentriert, in denen der Lernfortschritt am langsamsten erfolgt. Der Lernende hat damit den Eindruck, von seinem Partner individuell betreut zu werden, ohne durch irgendwelche Manipulationen von dem Lernziel abgelenkt zu werden.

Die Kopplung zwischen Studio und Scenario erfolgt absichtlich nur sehr lose, um auf der Scenarioseite sowohl andere Lernstoffe wie auch neue Entwicklungen auf dem audiovisuellen Bereich ohne Schwierigkeiten in das System integrieren zu können. Die Funktion des Studios kann dabei völlig frei durch Tonbandkassetten, Videorecorder, Multimedia-Shows oder einfach Fernsehsendungen wahrgenommen werden.

## Gestalten:

Mit der Möglichkeit der Farbwiedergabe, Grafik und programmierbaren Tonerzeugung auf der Basis des Personal Computers entsteht die Idee eines universellen neuen künstlerischen Instrumentes, mit dem sich statische wie auch bewegte Bilder, Musik und Lyrik hervorbringen lassen.

Um die sich bietenden Möglichkeiten weiten Kreisen zugänglich zu machen, müssen Systeme von Bedienungsanweisungen höchster Einfachheit und Flexibilität geschaffen werden, die sowohl technische wie auch sinnespsychologische Momente miteinander verbinden.

Insbesondere die Echtzeitproduktion bewegter Bilder, die in der klassischen Kunst nicht bekannt ist, erfordert Entscheidungen über prinzipielle Generierungsmodalitäten, die – nicht nur im Kunstbereich – richtungsweisend sein könnten. Dies gilt beispielsweise für die „grafische Improvisation", die mit solchen Geräten möglich ist. Als beachtenswert unkonventionelles Generierungsprinzip erweist sich eine Art laufenden Bildaufbaus mit vorgegebener Taktfrequenz – wahlweise unter Einbeziehung von Zufallsgeneratoren –, bei der der Benutzer in einer interaktiven Arbeitsweise die Möglichkeit des Eingriffes in die laufenden Prozesse hat.

Dabei handelt es sich im wesentlichen um die Veränderung von Parametern oder um den Wechsel von einer Aufbauroutine zur anderen. Den Schritt der elektronischen Zwischenrealisierung der grafischen Prozesse und einer Sammlung von Grundfunktionen, auf denen der Benutzer „spielen kann, wie auf einem Instrument", übernimmt dabei wieder ein spezielles Steuermodul für den Rechner.

Diese Methode bietet auch den Übergang zu zwei weiteren Generierungsstrategien: Einerseits kann der taktweise Aufbau an jeder Stelle unterbrochen werden; dadurch ist es möglich, den Prozeß auf die Produktion eines optimalen Endresultates einzurichten. Andererseits kann der zunächst dem Benutzer überlassene steuernde Eingriff auch voll an Zufallsgeneratoren übertragen werden, so daß sich ein automatischer Ablauf ergibt, der im Sinne eines filmischen Ablaufes auch als meditative Präsentation geeignet ist.

Diese Möglichkeiten, neuartige und unendlich vielfältige Impressionen zu erzeugen, müssen besonders diejenigen unter uns ansprechen, die es reizt, den Computer als neues Ausdrucksmittel ihrer eigenen Kreativität kennenzulernen.

Vor nicht sehr langer Zeit waren auch die preiswertesten Computer noch so teuer, daß selbst Firmenleitungen vor einer Anschaffung zurückschreckten. Seit zwei, drei Jahren aber sind in den deutschen Sprachgebrauch solche Begriffe wie „Heim-Computer" oder „Hobby-Computer" eingegangen. Diese Namen und vor allem die nun immer häufiger aufkommende Bezeichnung "Personal Computer" deuten darauf hin, daß sich Preissituation und Zielgruppen entscheidend geändert haben.

H. Schumny

# Personal Computer

Auch nach sorgfältiger Abwägung schien es nicht möglich, für diesen Beitrag einen deutschen bzw. „eingedeutschten" Titel zu finden – es sei denn, man versteht die Bezeichnung "Personal Computer" auch schon als eingedeutscht. Was wir darunter verstehen wollen, ist an anderen Stellen in diesem Jahrbuch schon angedeutet und auch mit *Home Computer* oder *Hobby Computer* beschrieben, nämlich komplette und einfach zu handhabende Mikrocomputer für viele verschiedene Anwendungsfälle im Heim- und Hobby-Bereich.

Nun ist aber leicht einzusehen und nachzuprüfen, daß „Hobby-Computer" eigentlich viel mehr können, als nur „Mondlandung" oder „Black Jack" zu spielen. Sie sind sicher ebenso für Kontenführungen oder Abrechnungen einsetzbar, mithin auch als *"Business Computer"* verwendbar. Auch bei einfacheren Datenerfassungs- und Steuerungsaufgaben in Industrie und Wissenschaft sind die gleichen „Hobby-Computer" als Kontrollrechner (*Controller*) oder vollständige Verarbeitungseinheiten geeignet.

So wird plausibel, daß all diese Bezeichnungen nur bestimmte Anwendungsfälle meinen können und nicht zur Beschreibung der mit den Computern gegebenen Möglichkeiten geeignet sind. Sie sind nämlich fast so etwas wie Universalcomputer, weshalb nun auch manchmal der Name „Mehrzweckcomputer" zu hören ist. Viel öfters wird aber in diesem Zusammenhang die englische Bezeichnung *General Purpose Computer* (GPC) gebraucht.

Eines darf natürlich nicht verschwiegen werden: Die Computer, von denen hier die Rede ist, sind in der Leistungsfähigkeit längst nicht mit Prozeßrechnern oder gar Großrechnern gleichzusetzen. Wenn auch ständig bessere Mikroprozessoren und erweiterte $\mu$C-Systeme vorgestellt werden, gibt es doch keinen Zweifel daran, daß zum Teil beträchtliche „Klassenunterschiede" bestehen bleiben; denn auch die Hersteller größerer und schnellerer DV-Systeme entwickeln ja weiter. Daraus läßt sich schliessen: Auch die Benennung "General Purpose Computer" beschreibt nicht hinreichend präzise solche Mikrocomputer wie z.B. den PET oder den Tandy TRS 80.

Wahrscheinlich muß man eher beim Preis ansetzen: Ein moderner Prozeßrechner kostet mit Standardperipherie vielleicht 100 000,– Mark. PET oder TRS 80 kosten zwei- bis dreitausend Mark. Das beantwortet eigentlich alles: Einen teuren Prozeßrechner kann sich nicht jeder leisten, einen PET schon eher. Das könnte der Grund dafür sein, daß sich nun die Bezeichnung *Personal Computer* durchsetzt. Es wird damit jedenfalls betont, daß von den produzierten Stückzahlen und vom Preis her diese Geräte zum „persönlichen Gebrauch" geeignet sind, wobei sich die erstaunlich niedrigen Preise ja gerade aus den hohen Stückzahlen ergeben.

Mag auch aus der Preissituation die Zielrichtung der Hersteller erklärt werden, nämlich jedermann zum eigenen Computer zu verhelfen; es gibt aber auch äußerliche und technische Merkmale, die den

*Personal Computers* gemeinsam sind. Sie sollen im folgenden angegeben und an den derzeit wichtigsten Vertretern dieser Rechnerklasse gegenübergestellt werden.

Typische Ausstattung eines *Personal Computers:*

- Tastatur wie bei Schreibmaschine oder ganz ähnlich (ASCII-Tastatur), oft ergänzt durch spezielle Steuer- oder Sondertasten;
- Bildschirm oder Anschluß für gewöhnlichen Fernseher vorhanden;
- Programmierung in BASIC;
- Verwendung von „Musikrecordern" zur Abspeicherung von Programmen – entweder eingebaut oder direkt anschließbar;
- Anschluß eines preiswerten Druckers möglich;
- diverse Erweiterungsmöglichkeiten käuflich;
- fertige Programme für verschiedene Anwendungsfälle bei den Computer-Herstellern oder über Benutzerclubs erhältlich.

Die Entwicklung zum "Personal Computer" begann etwa 1975. Hier im Zusammenhang noch einmal ein paar Angaben zur „µP-Geschichte":

1969 – Entwicklung der ersten „programmierbaren" Schaltkreise;

1971 – Auslieferung des 4bit-Mikroprozessors 4004;

1972 – 8bit-Prozessor 8008;

1973 – Auslieferung des µP 8080;

1974 – Verbesserte Version 8080A;

1975 – Vorstellung des Mikrocomputers ALTAIR 8800 von MITS, New Mexico.

Der ALTAIR 8800 enthielt den Prozessor 8080A und kostete etwa 2000 Dollar. Adressen und Daten mußten über 16 bzw. 8 Binärschalter eingegeben werden. Es folgten in den USA sehr schnell andere, verbesserte Mikrocomputer; Benutzerclubs und Shops wurden gegründet. Der internationale Durchbruch in Richtung "Personal Computer" gelang aber erst 1977/78 mit der weltweiten Auslieferung des PET 2001 von *Commodore* zum Preise von zunächst etwa 3000 Mark. PET steht für *Personal Electronic Transactor. Transact* kann etwa mit „erledigen" oder „abwickeln" übersetzt werden. Der PET wurde mithin als „elektronisches Werkzeug" angeboten, das allerlei persönliche Aufgaben abwickeln soll. Dieser PET wies alle Merkmale auf, die links als typische Ausstattung für Personal Computer aufgelistet sind. Mit vergleichbaren technischen Daten und Preisen sind inzwischen sehr viele Computer am Markt, wobei am meisten natürlich aus den USA kommt. Aber auch deutsche und japanische Hersteller sind in zunehmendem Maße dabei.

Abschließend sind im folgenden ein paar der Ende 1979 wichtigsten *Personal Computers* aufgelistet, wobei nach Herstellern alphabetisch geordnet wurde. Mit ziemlicher Sicherheit sind inzwischen weitere Computer bekannt und auch in der Bundesrepublik zu kaufen. Aber das ist ja gerade typisch für diesen Bereich, daß zum Zeitpunkt des Publizierens die technische Entwicklung schon wieder weitergegangen ist.

| Hersteller | Personal Computer | Prozessor |
|---|---|---|
| Apple | Apple II | 6502 |
| ATARI | ATARI 800 | 6502 |
| Commodore | PET 2001 | 6502 |
| | CBM 3001 | 6502 |
| Compucolor | Compucolor II | 8080 A |
| Elrad | Triton | 8080 A |
| Exidy | Sorcerer | Z80 |
| Kontron | PSI-80 | Z80 |
| Luxor | ABC 80 | Z80 |
| North Star | Horizon | Z80 |
| Ohio Scientific | Challenger | 6502 |
| Rockwell | AIM 65 | 6502 |
| Sanyo | SHC 1000 (Mopsy) | 6800 |
| Sharp | BA MZ-80K | Z80 |
| Siemens | PC 100 | 6502 |
| SKS/EMK | KISS | 8085 A |
| Tandy | TRS 80 | Z80 |

Durch den zunehmenden Einsatz von Mikroprozessoren wird für die Anwender der Bedarf an geeigneten Mikroprozessor-Entwicklungssystemen immer größer. Von den Halbleiterherstellern, aber auch von Meß- und Laborgeräteherstellern, werden eine Vielzahl solcher Entwicklungssysteme auf dem Markt angeboten. Da sich unter dem Begriff „Entwicklungssystem" Geräte verschiedenster Konzeption und Ausstattung verbergen, sollen im Rahmen dieses Beitrages grundlegende Unterschiede und Begriffe erläutert werden, und anhand einiger Ausdrucke soll dargestellt werden, welche Werkzeuge dem Anwender von Entwicklungssystemen bei Programmerstellung und Hardware/Software-Integration zur Verfügung stehen.

W. Handke

# Mikroprozessoren – Hilfsmittel zur Programmentwicklung und Testmethoden

## 1 Einteilung von Mikroprozessoren

Seit Anfang der 70er Jahre befinden sich Bauelemente der Digitaltechnik auf dem Markt, die zusammen mit anderen geeigneten Baugruppen dazu bestimmt sind, von Anwendern gestellte Aufgaben zu lösen. Die Besonderheit gegenüber den bekannten Lösungen mit verdrahteter Logik ist, daß die Lösung verschiedenster Aufgaben mit dem gleichen Schaltungsaufbau realisierbar ist. Dies wird dadurch erreicht, daß innerhalb eines solchen Bauelements bzw. einer solchen Baugruppe Abläufe durchgeführt werden, die in einem Programm festgelegt sind. Diese Bausteine der digitalen Datenverarbeitung werden allgemein *Mikroprozessor* oder **Minicomputer** genannt. Da beide Begriffe nebeneinander benutzt werden, sollte man sich grundsätzlich die Frage stellen, wieviele Baugruppen bzw. Bauelemente benötigt werden, um einen arbeitsfähigen Rechner zu erhalten. Der Begriff *Rechner,* in diesem Zusammenhang eigentlich falsch, wird jedoch vielfach benutzt, da solche Minicomputer häufig als Rechner ausgebildet sind; dies bedeutet jedoch nur eine spezielle Ausnutzung des Rechenwerks und einer speziellen Interpretation der verarbeiteten Daten.

Zur Abgrenzung von Minicomputer und Mikroprozessor sollen zunächst die benötigten Baugruppen mit ihrer Funktion aufgezeigt werden; diese Baugruppen sind typisch für Mikroprozessoren bis zum Großcomputer. Sie kommen aber nicht immer in klarer Form vor und können vielfältigen Abweichungen unterliegen.

Da alle Abläufe im Prozessor synchron erfolgen, wird eine *Takteinrichtung* benötigt. In einigen Fällen werden besondere Anforderungen an den Takt gestellt, was aber meistens von der Technologie des Prozessors herrührt. Ein Punkt ist in bezug auf den Takt von ganz besonderer Bedeutung: Zwischen der **Taktfrequenz** und der **Befehls-** oder **Programmausführungszeit** besteht kein direkter Zusammenhang.

**Minicomputer:** Hier werden damit ganz allgemein „kleine", also vollständige EDV-Anordnungen bezeichnet, also z.B. auch die neuen *Einchip-µP,* die außer Bedienungs- und Anzeigeelementen alle notwendigen Komponenten enthalten.

**Taktfrequenz:** Auch Zyklus oder System-Clock. Gemeint ist das Zusammenwirken aller Computerteile mit einem festen Systemtakt (z.B. 2 MHz).

**Befehlsausführungszeit:** Immer ein Vielfaches der Maschinen-Zykluszeit (z.B. 3 Zyklen, d.h. 1,5 µs beim 2-MHz-Takt).

*Akkumulator und* **ALU** (Arithmetisch-Logische Einheit) stellen das Herz der Datenverarbeitungsanlage dar. Die hier abzählbaren Fähigkeiten aus der Palette der möglichen Verknüpfungen ermöglicht eine Aussage über den Wert eines Bauelements zur Datenverarbeitung. Es seien einige dieser Verknüpfungen aufgezeigt:

*Arithmetische Verknüpfungen:*

| | |
|---|---|
| Addition | Multiplikation |
| Subtraktion | Division |

*Logische Verknüpfungen:*

| | |
|---|---|
| UND | Exklusives ODER |
| ODER | Schiebe-Operationen |
| NICHT | Bitweise Manipulationen |

In enger Zusammenarbeit mit dem Akkumulator und der arithmetisch-logischen Einheit steht noch das **Zustandsregister.** Hier werden Aussagen über die Ergebnisse von Verknüpfungen getroffen. „Null", „Vorzeichen", „Übertrag" sollen hier als Auswahl genannt sein.

Eine weitere Baugruppe ist der **Programmzähler.** Er zeigt auf den auszuführenden Befehl im Programmspeicher. **Inkrementiert** man den Programmzähler nach jeder Befehlsausführung, so erhält man ein geradeaus laufendes Programm. Die Möglichkeit den Programmzähler während eines Programms definiert zu stellen, bewirkt einen Programmsprung. Die Vielfalt der Sprungbefehle, insbesondere bedingter Sprünge (abhängig vom Inhalt des Zustandregisters), relativer Sprünge oder absoluter Sprünge, gibt ein gutes Bild über den Wert eines Prozessors.

Zum Speichern der Programme dienen in Mikroprozessor-Systemen meist **ROM**, zum weiteren stehen zum Speichern temporärer Daten **RAM** zur Verfügung. Ausgewählte Speicher (*Register*) sind eng an den Akkumulator geknüpft, können oft sehr einfach adressiert werden oder ihre Inhalte können direkt mit dem Akkumulatorinhalt verknüpft werden.

Unbedingt werden noch Eingabe/Ausgabe-Kanäle benötigt, die das System mit seiner Umwelt verbinden. Hier sind parallele und serielle Kanäle üblich; zur sicheren Datenübertragung dienen Steuerleitungen wie „Daten gültig" oder „Nicht zur Datenaufnahme bereit". Es gibt dabei eine Vielzahl von Steuermöglichkeiten, die alle unter dem Begriff **Handshake** vereint werden können.

Als letztes sei noch auf die Unterbrechungsmöglichkeiten einzugehen. Programmierte oder zufällige Unterbrechungen (**Interrupts**) sollen die Programmausführung unterbrechen und in gezielte Unterprogramme münden. Nach ihrer Ausführung soll mit dem Hauptprogramm fortgefahren werden. Der **Stackpointer,** ein spezielles Register, deutet auf eine Speicheradresse, an der die Unterbrechungsadresse sowie wichtige Prozessorinhalte gespeichert werden (*Stack*). Das ist nur eine der möglichen Interruptverarbeitungen. Jede der angebotenen Unterbrechungsverfahren haben ihre besonderen Vorteile wie Schnelligkeit oder beliebiges Verschachteln von Unterbrechungen.

Stellt man jetzt die möglichen Kombinationen der Baugruppen gegenüber, so findet man die heute gebräuchlichen Mikroprozessortypen:

### a) Allgemeiner Mikroprozessor (Bild 1)

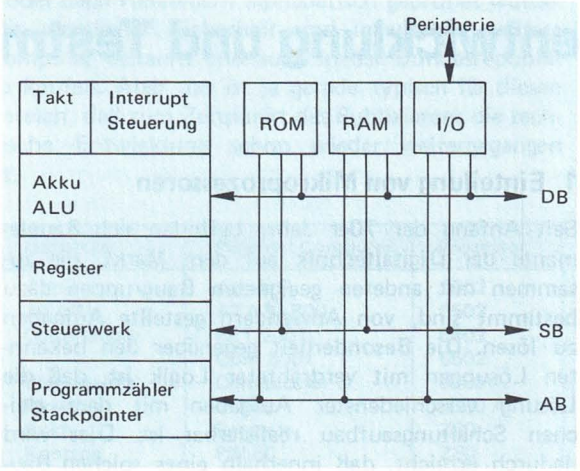



**Bild 1** Schema des allgemeinen $\mu$P (DB: Datenbus, SB: Steuerbus, AB: Adreßbus)

**ALU:** *Arithmetic Logic Unit;* Arithmetisch Logische Einheit, also das Rechenwerk in der CPU.

**Zustandsregister:** Enthält eine Reihe von Bits, die Auskunft über wichtige logische Zustände in der Zentraleinheit geben (*Flags, Statusbit, Statusregister, Prozessorstatuswort*).

**Programmzähler:** Enthält die Adresse des Befehls, der als nächster auszuführen ist (auch: Befehlszähler).

**Inkrementierung:** Erhöhung einer Adresse oder eines Datenwertes um eine Einheit (Dekrementierung: Verringerung um eins).

**RAM:** *Random Access Memory;* Schreib-Lesespeicher. Meist zur Abspeicherung von Daten und Ergebnissen benutzt.

**b) Einchip Mikroprozessor = Minicomputer (Bild 2)**

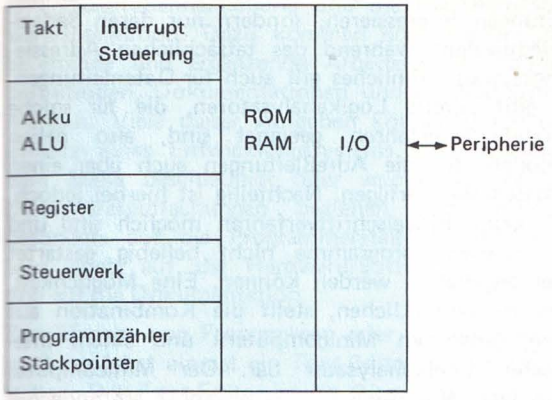



**Bild 2** Schema des Einchip-$\mu$P (I/O: Input/Output)

Die unter a) und b) genannten Prozessoren zeichnen sich durch folgende Eigenschaften aus:

Diskrete Datenbusbreite
Diskrete Adreßbusbreite
Feste Zuordnung von Befehl zu Ausführung

Im Gegensatz dazu stehen die Bit-Slice-Prozessoren, die durch ihre **Kaskadierbarkeit** beliebige Datenbreiten erreichen können. Ihr Steuerwerk (**Mikroprogrammwerk, MPM**) ist meist frei programmierbar, so daß auch die feste Zuordnung von Befehl zu Ausführung nicht mehr gegeben ist.

**c) Bit-Slice-Prozessoren (Bild 3)**

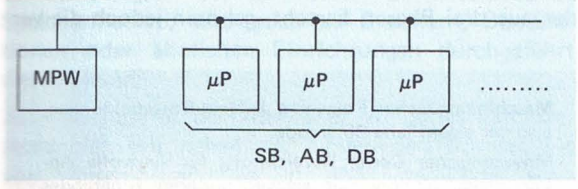



**Bild 3** Schema des Bit-Slice-Prozessors

## 2 Klassifikation von Mikroprozessor-Entwicklungssystemen

Wie bereits festgestellt wurde, liegt das Wesen einer Anlage zur Datenverarbeitung in ihrer Programmierung. Dies steht völlig im Gegensatz zur verdrahteten Logik, in die der Entwickler seine „Programme" schon bei der Entwurfsphase mit einbringt und nachträgliche Änderungen nahezu ausschließt. Mit einer einmal entworfenen und hergestellten Standardschaltung, in Abschnitt 1 als Mikroprozessor beschrieben, lassen sich eine Vielzahl von Problemen lösen. Hierzu ist es jedoch notwendig, die jeweiligen Mikroprozessoren genau zu kennen, um ein Programm zu erstellen.

Einen Mikroprozessor kennenzulernen heißt, zunächst einmal intensiv die Datenblätter zu studieren und seinen Befehlssatz zu begreifen. Eine bedeutende Hilfe bei dieser Arbeit stellen die sogenannten *Mikroprozessor-Kits* dar. Solche Kits sind meist eine minimale, aber oft ausbaubare, Mikroprozessor-Schaltung. Ergänzt wird diese Schaltung, um die Kommunikation mit dem Anwender herzustellen, mit einer Tastatur und einem Ausgabekanal zu einer Sieben-Segment-Anzeige oder einfachen LEDs. Diese Kits benötigen ein **Betriebssystem (Monitor)** zur Tastenabfrage und zur Anzeige eingegebener Programme. Oft sind auch Hilfsmittel zur Fehlerbeseitigung (**Debug**) vorhanden, die es ermöglichen, Programme in Einzelschrittverfahren abzuarbeiten und eventuell sogar Register im Prozessor zur Anzeige zu bringen. Selbst wenn zusätzliche Anschluß-

**ROM:** *Read Only Memory;* Festwertspeicher. Kann nur gelesen werden; meist zur Abspeicherung von fertigen (ausgetesten) Programmen benutzt.

**PROM:** Programmierbarer Festwertspeicher.

**Handshake:** (Händedruck). Bezeichnet werden damit Signale, die eine Rückmeldung, Empfangsbestätigung, Fertigmeldung etc. darstellen.

**Interrupt:** Gewollte oder ungewollte Programmunterbrechung. Meist genutzt, um spontan auftretende Ereignisse sofort verarbeiten zu können (s. auch Beitrag „Interrupts in Mikroprozessor-Systemen" von *P. Welzel*).

**Stackpointer:** Spezielles Register, das angibt, welche Speicherstelle im Stapelbereich (*Stack*) als nächste belegt werden kann.

**Kaskadierung:** Zusammenschaltung einzelner Prozessoren mit beispielsweise 4bit Wortbreite zu einem Prozessor großer Wortbreite (z.B. 16 bit).

**Mikroprogrammwerk:** Funktionseinheit mit Festwertspeicher (ROM), in dem Programme stehen, die den Befehlsablauf steuern.

**Betriebssystem:** Programme, die den „Betrieb" einer DV-Maschine ermöglichen (vom Hersteller fest eingespeichert).

**Monitor:** Teile eines Betriebssystem, die das Einspeichern von Daten und das Anzeigen von Adressen und darin enthaltenen Daten ermöglichen.

**Debugging:** Suchen und Beseitigen von Programmierfehlern.

möglichkeiten zu Kassettenrecordern bestehen, ist die Erstellung längerer Programme äußerst mühsam. Einerseits wird in **Maschinensprache** programmiert zum andern wird das Programm sofort adreßrichtig im Speicher (RAM) abgelegt. Hinzufügen von Programmteilen ist deshalb nur mit komplizierten Monitorprogrammen möglich. Zusätzliche Hilfsmittel, wie Programmdokumentation oder sogar Ausdrucke mit Kommentaren, werden nahezu unmöglich. Deshalb sei zusammenfassend bemerkt, daß Mikroprozessor-Kits nur zum generellen Kennenlernen von Prozessor und Programmiertechnik wertvoll sind, ein tatsächliches Werkzeug zur Programmerstellung jedoch nicht darstellen.

Eine bedeutende Hilfe wäre es, wenn man im **mnemonischen Code** (*Assembler*) des Prozessors Programme schreiben könnte und die Maschine den Übersetzungsvorgang durchführen könnte. Hierfür eignet sich jeder Minicomputer, der über eine Eingabetastatur verfügt und eine geeignete Ausgabevorrichtung besitzt. Die Eingabetastatur muß jetzt schon, im Gegensatz zum Kit, über den gesamten Zeichensatz (Zahlen, Buchstaben, Sonderzeichen) verfügen, um eine Programmeingabe in der definierten Assemblersprache zu ermöglichen. Als Ausgabe geeignet wären entweder Datenträger wie Lochstreifen, Band oder Diskette; aber auch Ausdrucke zum nachträglichen Übertragen in die Mikroprozessorschaltung. Man könnte auch das Programm direkt in die PROMs „brennen", was aber dann nachteilig ist, wenn das Programm noch Fehler enthält.

Besitzt der Minicomputer neben dem Übersetzungsprogramm noch die Möglichkeit, Programme für einen Mikroprozessor zu simulieren, so kann man jetzt schon von einem *Mikroprozessor-Entwicklungssystem* sprechen. Einen Mikroprozessor auf einem anderen Prozessor zu simulieren, stößt jedoch auf einige Schwierigkeiten. So ist es zu erklären, daß viele Minicomputer für die Programmentwicklung eines Mikroprozessors oder einer Mikroprozessorfamilie ausgelegt sind und die Anwenderprogramme direkt auf der CPU des Minicomputers ausführen.

Wie aber eingangs schon beschrieben, stellt sich neben der Programmierung auch die Aufgabe, die Hardware einer Mikroprozessorschaltung aufzubauen und auszutesten. Mit den Hilfsmitteln der üblichen Meßtechnik stößt man sehr bald an Grenzen. Ein Ausweg sind die **Logikanalysatoren.** Hier werden allerdings eine Vielzahl von Kanälen benötigt, um z.B. Adreßleitungen und Datenleitungen darzustellen. Bei den üblichen Mikroprozessoren kommt man hier leicht auf 24 bis 32 Kanäle. Nachteilig ist dabei außerdem, daß von den 16 Adreßleitungen eigentlich meist nicht die logischen Zustände der einzelnen Leitungen interessieren, sondern nur deren **Sedezimaläquivalent** während des tatsächlichen Adressierungszyklus. Ähnliches gilt auch für Datenleitungen. Es gibt bereits Logikanalysatoren, die für solche Darstellungsverfahren geeignet sind, also neben Decodern für die Adreßleitungen auch über einen **Disassembler** verfügen. Nachteilig ist hierbei jedoch, daß keine Einzelschrittverfahren möglich sind und die Anwenderprogramme nicht beliebig gestartet oder angehalten werden können. Eine Möglichkeit, dies zu verwirklichen, stellt die Kombination aus oben genannten Minicomputern und einem integrierten Logikanalysator dar. Der Minicomputer muß jetzt aber auch für die bekannte Methode der **'In Circuit Emulation'** ausgerüstet sein. *Emulation* heißt in diesem Zusammenhang Nachbildung des Prototypen-Mikroprozessors durch die CPU des „Minicomputers". Hier sieht man, daß der Minicomputer nur noch für die Hardware/Softwareintegration für einen Mikroprozessortyp herangezogen werden kann. Nachteilig ist bei diesem Verfahren weiterhin, daß sich die Speicherbereiche der Anwenderprogramme und Betriebsprogramme wie Monitor oder Debug-Software nicht überlappen dürfen.

## 3 Anforderungen an ein zukunftsorientiertes Entwicklungssystem

Bei der Klassifikation von Mikroprozessor-Entwicklungssystemen wurden schon einige Anforderungen genannt. Zu einem vollständigen Entwicklungszyklus, der aus drei Phasen besteht, gehören jedoch die ver-

**Maschinensprache:** Folge von Ja-Nein-Entscheidungen, also der eigentliche Binärcode.

**Mnemonischer Code:** Bezeichnung für sinnvolle Abkürzungen für die nur schwer lesbaren Binärcodes (auch: *Assemblersprache*).

**Logikanalysator:** Meßgerät, das die logischen Zustände (Ein oder Aus) auf Leitungen anzeigt, also z.B. die Datenmuster (*Binärcodes*) auf einem Datenbus.

**Sedezimaläquivalent:** Abgekürzte Darstellung von Binärcodes, wobei jeweils 4 Binärstellen durch eine Hexadezimalzahl beschrieben werden.

**Disassembler:** Übersetzungsprogramm, das die zu den Binärcodes (Maschinensprache) gehörenden mnemonischen Ausdrücke darstellt.

**In Circuit Emulation:** Vollständige Nachbildung (*Emulation*) eines μP innerhalb einer Schaltung.

schiedensten Werkzeuge. Die drei Phasen sind Programmentwurf und Erstellung, Schaltungsentwurf und Prototypenherstellung und die Hardware/Software-Integration. Dazu kommen noch weitere vorbereitende und ergänzende Arbeiten wie Produktdefinitionen, Dokumentationen und Produktabstimmungen. Viele dieser Aufgaben können durch den Einsatz eines Entwicklungssystems in entscheidendem Maße beschleunigt oder vereinfacht werden. Diese Vereinfachungen beziehen sich natürlich besonders auf die Programmerstellung und Tests, aber auch auf die Hardware/Software-Integration und auf die Dokumentation.

Zur Eingabe von Programmen oder anderen Texten muß zunächst einmal ein *Text*-**Editor** zur Verfügung stehen. Der Text-Editor ist ein *Dienstprogramm,* das auf der System-CPU läuft, während gleichzeitig ein Speicherbereich als Arbeitsspeicher zur Texteingabe benutzt wird. Daneben besteht auch die Möglichkeit, die Editierungseigenschaften intelligenter Terminals auszunutzen und fertig erstellte „Seiten" als Ganzes zu den Speichermedien des Entwicklungssystems zu senden. Ein wesentlicher Unterschied liegt in zeilen- und zeichenorientierten Editoren. In einem Fall werden gesamte Zeilen aufgerufen, um Änderungen vorzunehmen, im anderen Fall können direkt mit dem **Cursor** Textzeichen geändert, eingefügt oder gelöscht werden. Ein Text-Editor ist dann als komfortabel anzusehen, wenn neben der Möglichkeit, ganze Datensätze zum Speichermedium zu transferieren, auch Teilmengen des Datensatzes zum oder vom Speichermedium transferiert werden können. Daneben sollte ein Editor danach beurteilt werden, ob die Ausgabe des Datensatzes zu verschiedenen Speichermedien, also auch zu Druckern, Bandstationen oder ähnlichen Einrichtungen durchgeführt werden kann.

Als nächstes sei der *Assembler* zu betrachten. Der Assembler soll neben der eindeutigen *Übersetzung von Assemblersprache in Maschinen-Code* auch eine umfassende Fehlererkennung und Meldung durchführen. Weiterhin müssen Direktiven enthalten sein, die eine übersichtliche Programmerstellung und Dokumentation gestatten. Unter Makro-Assemblern versteht man solche, die einmal beschriebene **Makros** beliebig oft in den Programmfluß einbinden können. Besonders vorteilhaft ist hier die bedingte Makro-Assemblierung. Hierzu muß aber vorausgesetzt werden, daß durch **String-Manipulationen** geeignete Abfragebedingungen geschaffen werden können. Die Geschwindigkeit des Assembliervorganges wird wesentlich von der Zugriffszeit zu den Speichermedien bestimmt, weshalb hier Diskettenspeicher besonders vorteilhaft sind. Assembler können grundsätzlich zweierlei Codes generieren, nämlich erstens solchen, der auf einer absolut angegebenen Adresse basiert oder zweitens einen Code, der in einem weiteren Arbeitsgang, dem **Link-Prozeß,** an eine beliebig angegebene Adresse verschoben werden kann. Bei Benutzung eines Linkers wird sich besonders vorteilhaft herausstellen, daß sich bereits fertige und ausgetestete Programmteile beliebig aus Programmbibliotheken in das Programm einbinden lassen.

Die Verwendung von Hochsprachen im Zusammenhang mit Mikroprozessorsystemen ist strittig. Meist wird eine hohe Speicherausbaustufe an das System selbst gestellt; außerdem ist die Erstellung effizienten Maschinen-Codes dann fraglich, wenn der Übersetzer nicht mit guten Optimierungsmethoden arbeitet. Praktisch ist, wenn der Übersetzer zunächst nur in die Assemblerebene übersetzt. Damit kann der Anwender wesentlich zur Optimierung beitragen, indem er nur Programmteile übernimmt, andere in Assembler-Sprache programmiert.

Die bis jetzt beschriebenen Methoden der Programmerstellung können nur **syntaktische Fehler** erkennen, nicht jedoch *logische Fehler* innerhalb eines Programms. Jetzt kommt die eigentliche Stärke eines Mikroprozessorentwicklungssystems erst zum Tragen, denn bis hierhin hatte sich ja das Entwicklungssystem noch nicht von einem Minicomputer unterschieden. Entwicklungssysteme bieten somit eine Möglichkeit, Mikroprozessor-Programme ablaufen zu lassen und dabei den Ablauf für den Entwickler darzustellen. Dazu muß das System selbst mit einem Prozessor und der nötigen *Peripherie* ausgerüstet sein. An dieser Stelle zeigt sich auch,

**Editor:** Programm, das das Schreiben, Ändern und Speichern von Programmen unterstützt.

**Cursor:** (Positionsanzeiger). Punkt oder Strich, der auf einem Bildschirm anzeigt, an welcher Stelle geschrieben oder gelöscht werden kann.

**Makro:** Ein oft gebrauchter Programmteil kann unter einem typischen „Namen" abgelegt und dann beliebig oft in ein Programm „gebunden" werden.

**String:** Zeichenkette, die als Einheit aufgefaßt wird.

**Link-Prozeß:** Durch *Linking* (Anbindung) können verschiedene Programmteile aneinandergereiht werden.

**Syntaktischer Fehler:** Verletzung der festgelegten Schreibregeln, also z.B. falscher Buchstabe oder Komma statt Punkt.

daß universelle Mikroprozessorentwicklungssysteme mit verschiedenen Prozessoren, nämlich genau den nachzubildenden, ausgerüstet werden können. Diese Forderung wird dann noch offensichtlicher, wenn in der Hardware/Software-Integrationsphase der Prozessor der Prototypenschaltung durch das Entwicklungssystem ersetzt wird. Daß in der Integrationsphase noch andere Forderungen gestellt werden können, ist selbstverständlich. *Einzelschrittverfahren,* **Tracing, Breakpoints,** *Historiendarstellung, Disassemblierung* und *Echtzeitanalyse* mit anschließender Darstellung sind hier die Schlagworte. Aber gerade hier kann ein Entwicklungssystem seinen Wert besser beweisen, als in anderen Betriebsarten wie z.B. **Edit Mode.**

Ergänzend noch ein Blick auf das Gesamtsystem:
Fehlen sollten niemals *Normschnittstellen zur Peripherie.* Hier werden neben den gebräuchlichen V-24-Schnittstellen und 20-mA-Stromschleifen auch Parallelschnittstellen angeboten. Bedenken sollte man, daß neben dem physikalischen Vorhandensein einer Schnittstelle auch die Software-Treiber vorhanden sein müssen, damit eine Schnittstelle universell gestaltet werden kann. Erwähnt seien nur wählbare **Parität,** *Anzahl von Start/Stopbits, Datenformate* u.a.m. Besondere Aufmerksamkeit ist der Tatsache zu schenken, ob ein Mikroprozessorentwicklungssystem an Gastrechner angeschlossen werden kann, da dadurch die Tür zu jedweder Programmierhilfe aber auch zum konzernweiten Datentransfer (**Modemverbund**) aufgestoßen wird.

Angesprochen wurden bereits die *Massenspeicher.* Diskettenspeichern wird hier im allgemeinen den Vorzug gegeben. Die relativ schnelle Zugriffszeit ist hier mit einer großen Datendichte vereint, die weder von Bandspeichern oder gar Lochstreifen erreicht wird. Hard-Discs würden die oben angesprochenen Vorteile noch überbieten, sind mir aber im Zusammenhang mit Entwicklungssystemen unbekannt. Bei geeigneten Verbindungsmöglichkeiten zu Gastrechnern ist diese Art der Datenspeicherung nicht ausgeschlossen.

Bei der technischen Realisierung von Mikroprozessor-Entwicklungssystemen ist eigentlich nur von Interesse, wo die Anwenderprogramme (Maschinen-Codes) während der Programmausführung abgelegt werden und wo die im Debug benötigten Dienstprogramme abgelegt sind und ausgeführt werden. Vor der Diskussion der Zusammenhänge zeigen vereinfachte Schemata eine Auswahl aus möglichen Realisierungen (**Bild 4**).

In **Bild 4a)** ist System- und Anwender-Prozessor (Emulator) identisch. Ein solches System ist somit nur für die Emulation eines Mikroprozessors zu verwenden. Zu bemerken ist außerdem, daß neben dem **Initialisierungs-ROM** auch bestimmte Debug-Programmteile geladen sein müssen, daß heißt, daß für die Anwenderprogramme nie der gesamte adressierbare Speicherbereich zur Verfügung steht. Der Einwand, daß Anwenderprogramme selten den gesamten Speicherbereich, z.B. 64 K-Byte belegen, wird dadurch widerlegt, daß vielfach wegen einfacherer Adressierung eine Programmverteilung auf den gesamten Adreßbereich gewählt wird.

Die in **Bild 4b)** gezeigte Maschine kann durch Einsetzen verschiedener Anwender-Prozessoren für die Emulation mehrerer Mikroprozessoren verwendet werden, ist aber sonst mit den Nachteilen des unter **Bild 4a)** beschriebenen Systems behaftet.

Anders ist dies bei der unter **Bild 4c)** gezeigten Maschine. Wahlweise eingesetzten Emulator-Prozessoren steht ein Speicher zur Verfügung, der den gesamten adressierbaren Bereich überdeckt. Betriebsprogramme sind genau wie Initialisierungs-ROM der Systemseite zugeordnet. Eine geeignete Interrupt-Struktur und BUS-Schalter können in bestimmten Betriebszuständen Anwender- und Systemseite miteinander verbinden. Dies ist z.B. beim Laden von Anwenderprogrammen vom Massenspeicher ins anwenderseitige RAM der Fall.

Das TEKTRONIX Mikroprozessor-Entwicklungssystem 'MDA 8002A' ist nach dem unter **Bild 4c)** beschriebenen Prinzip aufgebaut. Damit sind alle Vorteile dieser Systemstruktur verfügbar; angefangen mit der Emulation für derzeit neun Mikroprozessortypen, über volle Speicherbereitstellung für den

**Tracing:** Schrittweise Ausführung und Protokollierung eines Programms.

**Breakpoint:** Programmierter Unterbrechungspunkt, um bei der Programmentwicklung Fehlersuchen zu erleichtern.

**Edit Mode:** Ablaufphase, während der an verschiedenen Stellen abgespeicherte Programmteile logisch richtig zusammengebunden werden.

**Parität:** Ergänzung der Einsbit eines Datenwortes auf eine festgelegte (gerade oder ungerade) Anzahl (Parität) zur Ermittlung von Übertragungsfehlern.

**Modem:** Einrichtung zur Modulation bzw. Demodulation bei der Datenfernübertragung.

**Initialisierung:** Ermöglichen und Starten von Grundoperationen.

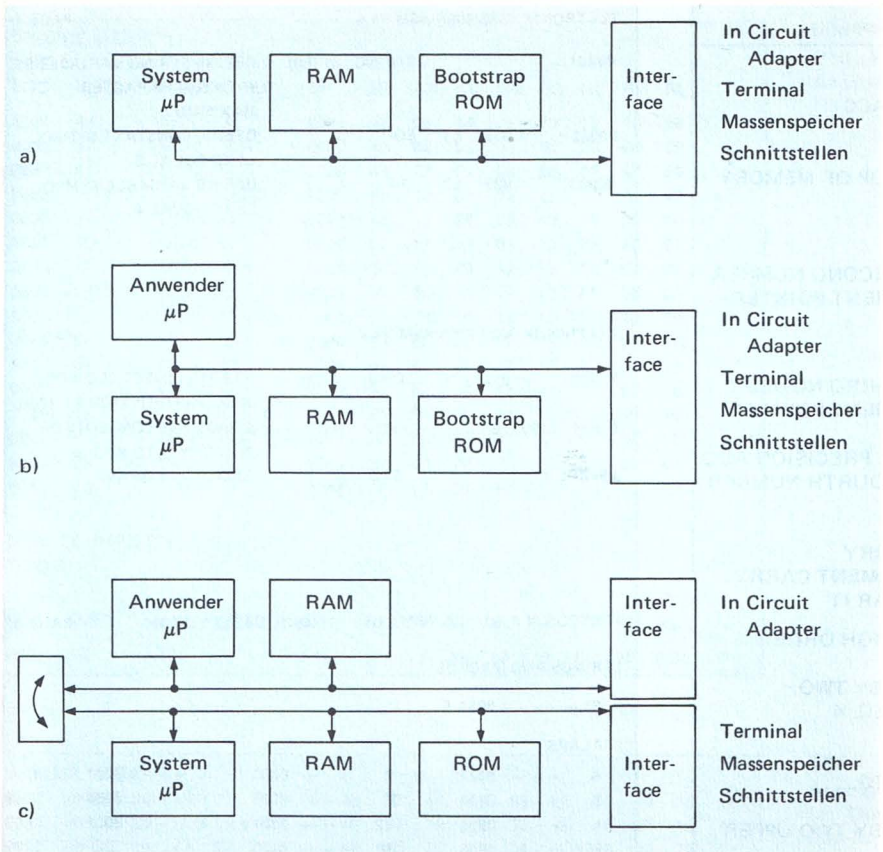



**Bild 4**
Ausführungsmöglichkeiten für Entwicklungssysteme

Emulator, bis hin zum Echtzeitbetrieb mit dem eingebauten Logikanalysator. Hinzu kommt, daß für Assemblierung und Compiler-Dienstprogramm noch ein weiterer Prozessor bereitsteht, der für diese Aufgaben besonders geeignet ist. Im folgenden Abschnitt sollen einige Darstellungsweisen vom Programmentwurf bis zum Echtzeitbetrieb gezeigt werden. Neben den Darstellungen des 'MDA 8002A' werden weitere Möglichkeiten diskutiert.

## 4 Darstellungsweisen am Systemterminal beim Programmentwurf

Es kann davon ausgegangen werden, daß bei den üblichen Entwicklungssystemen jede Kommunikation mit dem Anwender über das **Systemterminal** erfolgt. In vielen Fällen gibt es am Entwicklungssystem selbst nur einen **'Master Reset'** und eventuell einige wenige Leuchtdioden-Anzeigen, die aber nur die nötigsten Betriebszustände anzeigen sollten. Je mehr diese Betriebszustände am Systemterminal abzufragen sind oder selbsttätig angezeigt werden, desto höher ist der Wert eines Entwicklungssystems einzuschätzen. In den einzelnen Betriebsarten, wie Editor, Assembler und Debug sollten klar strukturierte Anzeigen Eindrücke vom momentanen Geschehen bzw. Ergebnisse dargestellt werden. Die Ausdrücke oder Anzeigen an einem Video-Terminal, das übrigens wegen seiner größeren Geschwindigkeit und geringeren Betriebskosten einem druckenden Terminal vorzuziehen ist, sollten im Edit-Mode genau

**Systemterminal:** Ein-Ausgabestation (z.B. Sichtgerät mit Tastatur) zum Bedienen einer EDV-Anlage.

**Master Reset:** Festverdrahteter und auf einen Schalter gelegter Befehl, mit dem per Interrupt der Prozessor in einen definierten Anfangszustand versetzt werden kann.

| Befehle | | Kommentare |
|---|---|---|
| ORG | ØØ | |
| XRA | A | ;CLEAR ACC |
| MOV | B, A | |
| MOV | H, A | |
| LXI | D, 13FFH | ;LOAD TOP OF MEMORY |
| LDAX | D | |
| DCX | D | |
| MOV | C, A | |
| LDAX | D | ;LOAD SECOND NUMBER |
| DCX | D | ;DECREMENT POINTER |
| MOV | L, A | |
| DAD | B | |
| LDAX | D | ;LOAD THIRD NUMBER |
| DCX | D | ;DECREMENT POINTER |
| MOV | C, A | |
| DAD | B | ;DOUBLE PRECISION ADD |
| LDAX | D | ;LOAD FOURTH NUMBER |
| MOV | C, A | |
| DAD | B | |
| STC | | ;SET CARRY |
| CMC | | ;COMPLEMENT CARRY. I.E. CLEAR IT |
| MOV | A, H | ;MOVE HIGH ORDER BYTE |
| RAR | | ;DIVIDE BY TWO |
| MOV | H, A | ;SWAP REG. 4 |
| MOV | A, L | |
| RAR | | |
| MOV | L, A | ;SWAP REG. |
| MOV | A, H | |
| RAR | | ;DIVIDE BY TWO UPPER BYTE |
| MOV | A, L | ;LOAD LOWER BYTE |
| RAR | | ;DIVIDE BY TWO ANSWER IN ACC. |

**Bild 5** Beispiel für ein Quellenprogramm

```
TEKTRONIX 8080/8085 ASM Vx.x                                  PAGE 1

ØØØØ1                 STRING  S1 (8Ø)   ;DEFINE STRING VARIABLE S1
                                        ;WITH 8Ø CHARACTER
                                        ;MAXIMUM
ØØØØ2      ØØØ3 L1    EQU     3         ;DEFINE CONSTANT SYMBOL
                                        ;L1 TO EQUAL 3
ØØØØ3      ØØØ4 L2    SET     4         ;DEFINE VARIABLE SYMBOL
                                        ;L2 TO EQUAL 4
.
.
.

TEKTRONIX 8080/8085 ASM Vx.x                                  PAGE 2

ØØØØ5      Ø1ØØ >     ORG     1ØØH      ;STARTS OBJECT CODE OF
                                        ;NEXT INSTRUCTION AT 1ØØH
ØØØØ6 Ø1ØØ 7E         MOV     A,(M)     ;LOADS THE CONTENTS OF
                                        ;MEMORY INTO REG. A
ØØØØ7                 END               ;END OF PROGRAM
.
.
.
.

TEKTRONIX 8080/8085 ASM Vx.x   SYMBOL TABLE LISTING           PAGE 3

STRINGS AND MACROS

  S1 - - - - - ØØ5Ø S

SCALARS

  A   - - -  ØØØ7     B   - - -  ØØØØ     C - - -  ØØØ1
  D   - - -  ØØØ2     E   - - -  ØØØ3     H - - -  ØØØ4
  L   - - -  ØØØ5     L2  - - -  ØØØ4V    M - - -  ØØØ6
  PSW - - -  ØØØ6     SP  - - -  ØØØ6

% (default) SECTION ØØØ1

  L1 - - - - - (Ø1ØØ)

7 SOURCE LINES     7 ASSEMBLED LINES     1ØØØ BYTES AVAILABLE
```

**Bild 6** Möglichkeiten der Dokumentation mit dem Entwicklungssystem MDA 8002 A

den eingegebenen Text in gleichem Format wiedergeben, wie das zur übersichtlichen Dokumentation nötig ist (**Bild 5**).

An die *Assembler* **Listings** werden schon höhere Anforderungen gestellt. Neben dem eingegebenen **Source Code** sollte der Maschinen-Code dargestellt werden. Hierdurch wird es auf einfache Weise möglich, die angezeigten Fehler zu interpretieren, sofern der Assembler über eine ausreichende Fehlermeldung verfügt. Folgender Ausdruck vom 'MDA 8002A' (**Bild 6**) zeigt ein Assembler Listing mit Adressen, Daten, Source und Kommentar. Eine abschließende Tabelle gibt Auskunft über verwendete Symbole, Stringnamen und im Falle von linkbaren Modulen über ungebundene Symbole. Eine zusätzliche Symbol-Referenz-Liste könnte das Assembler Listing optimal ergänzen.

**Listing:** Auflistung von Programmbefehlen.

**Source Code:** Die vom Programmierer benutzte Sprache (z.B. *Assembler*), die noch in die Maschinensprache übersetzt werden muß.

```
>TRACE ALL
>GO 0
LOC   INST    MNEM OPER    SP    RF  RA  RB  RC  RD  RE  RH  RL

0000  AF      XRA  A       0000  46  00  00  00  00  00  00  00
0001  47      MOV  B,A     0000  46  00  00  00  00  00  00  00
0002  67      MOV  H,A     0000  46  00  00  00  00  00  00  00
0003  11FF13  LXI  D,13FF  0000  46  00  00  00  13  FF  00  00
0006  1A      LDAX D       0000  46  12  00  00  13  FF  00  00
0007  1D      DCX  D       0000  92  12  00  00  13  FE  00  00
0008  4F      MOV  C,A     0000  92  12  00  12  13  FE  00  00
0009  1A      LDAX D       0000  92  0C  00  12  13  FE  00  00
000A  1D      DCX  D       0000  92  OC  00  12  13  FD  00  00
000B  6F      MOV  L,A     0000  92  0C  00  12  13  FD  00  0C
000C  09      DAD  B       0000  92  0C  00  12  13  FD  00  1E
000D  1A      LDAX D       0000  92  B1  00  12  13  FD  00  1E
000E  1D      DCX  D       0000  96  B1  00  12  13  FC  00  1E
000F  4F      MOV  C,A     0000  96  B1  00  B1  13  FC  00  1E
0010  09      DAD  B       0000  96  B1  00  B1  13  FC  00  CF
0011  1A      LDAX D       0000  96  54  00  B1  13  FC  00  CF

>TRACE JMP STEP
>GO 0

LOC   INST    MNEM OPER    SP    RF  RA  RB  RC  RD  RE  RH  RL

0024  FE      RST          FFFE  97  15  00  00  13  FF  00  2B
```

**Bild 7**
Vollständige Programmauflistung mit Angabe der Registerinhalte

```
> DUMP 7D B4
0070  =03  0C  19  18  E4  20  18  70  E4  0D  98  2E  0C  19  15  18
0080  =17  20  CC  19  15  0C  19  14  18  18  0E  19  4E  0F  19  4F
0090  =CE  19  4A  CF  19  4B  1B  4D  0C  19  14  1C  40  03  20  CC
00A0  =19  14  0E  19  4C  0F  19  4D  1B  66  E4  3A  98  0C  CC  19
00B0  =2B  0C  19  15  1C  40  65  1F  40  81  0C  19  19  1C  41  00

          MAP P 0000-7FFF  C000-FFFF

          MAP U FC00

          MAP U 8000-BFFF

0XXX  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
1XXX  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
2XXX  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
3XXX  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --

4XXX  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
5XXX  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
6XXX  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
7XXX  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --

8XXX  ..  ..  ..  ..  ..  ..  ..  ..  ..  ..  ..  ..  ..  ..  ..  ..
9XXX  ..  ..  ..  ..  ..  ..  ..  ..  ..  ..  ..  ..  ..  ..  ..  ..
AXXX  ..  ..  ..  ..  ..  ..  ..  ..  ..  ..  ..  ..  ..  ..  ..  ..
BXXX  ..  ..  ..  ..  ..  ..  ..  ..  ..  ..  ..  ..  ..  ..  ..  ..

CXXX  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
DXXX  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
EXXX  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
FXXX  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
```

**Bild 8**
Auszug ausgewählter Speicherbereiche (a: Dump) und Speicherbereichsdefinition (b: Map)

Wichtig für die Phase der Software-Tests sind die Möglichkeiten, die das Debug-Dienstprogramm dem Anwender eröffnet. Neben einem *Trace* der vollständigen Programmausführung (**Bild 7**) sollte die Möglichkeit bedingter Traces, z.B. bezogen auf Adreßbereiche oder Befehlstypen, bestehen. Es sollte auch definiert werden können, daß Anwenderprogramme an festgelegten Unterbrechungspunkten angehalten werden können und dann Angaben über die aktuellen Inhalte der Prozessorregister gemacht werden. Unterbrechungspunkte (*Breakpoints*), die zusätzlich von bestimmten Bedingungen abhängig

zu definieren sind, geben dem Anwender Mittel in die Hand, nahezu jeden Fehler innerhalb der erstellten Programme schnell zu lokalisieren. Einige Entwicklungssysteme sind in der Lage, die Vergangenheit (*History*) des Programmlaufs vor Erreichen des Breakpoint festzuhalten und zu dokumentieren.

Ergänzend zu diesen Methoden der Fehlersuche und -Beseitigung, muß auch der Inhalt von Speichern dargestellt werden können. Es sollen Änderungen in Speichern und Mikroprozessorregistern durchführbar sein und Speicherinhalte sollen verschoben werden können. Der Sinn solcher Manipulationen wird jedem Anwender bekannt sein und ist deshalb aus keinem Entwicklungssystem wegzudenken (**Bild 8**).

## 5 Werkzeuge zur Hardware/Software-Integration

Um die fertiggestellten und getesteten Programme zusammen mit der parallel dazu entworfenen und aufgebauten Prototypen-Schaltung zu prüfen, muß das Entwicklungssystem über die Fähigkeit der '*In Circuit-Emulation*' verfügen. Nach Einsetzen eines Tastkopfes anstelle des Mikroprozessors der Prototypen-Schaltung stehen dem Anwender wieder alle Mittel des Software-Debug zur Verfügung. Anwenderprogramme können an beliebigen Adressen gespeichert werden, Trace und Breakpoints können gewählt werden. Eine ergänzende Maßnahme ist das '**Memory Mapping**'. Damit werden Speicherbereiche definiert, die aus dem RAM des Entwicklungssystems zur Programmausführung benutzt werden, und andere, die aus der Prototypen-Schaltung angesprochen werden. Der Sinn ist es, bereits fertig ausgetestete Programmteile im PROMS zu „brennen" und in die Prototypen-Schaltung einzusetzen. Wünschenswert und äußerst effektiv wird diese "Mapping"-Methode, wenn relativ kleine Speicherbereiche definierbar sind.

Die gesamte Hardware-Testtechnik wird jedoch dann relativ nutzlos, wenn das Entwicklungssystem nicht in der Lage ist, Anwenderprogramme in Echtzeit auszuführen. Echtzeittests sind wertlos, wenn ausschließlich 'GO', 'NO GO' Aussagen getroffen werden können, da der Anwender im Falle von 'NO GO' ratlos vor seinem System sitzt. Abhilfe schafft hier nur, wenn Echtzeitaufzeichnungen während der Programmausführung erfaßt und nachträglich zur Anzeige gebracht werden können. Daß hier allerdings mit einem komfortablen Triggersystem gearbeitet werden muß, versteht sich von selbst. Da die beschriebene Erfassungstechnik jedoch mit kleinen Halbleiterspeichern auskommen soll, wird meist prozessorsynchron erfaßt, so daß zeitkritische Vorgänge wie verschliffene Impulsflanken oder '**Glitches**' nicht darstellbar sind. Für diesen Zweck kann man an einem Entwicklungssystem das Erreichen der **Triggerereignisse** als Triggerimpulse für gewöhnliche Speicheroszilloskope aussenden, die dann ein vollständiges Bild der kritischen Signale aufzeigen können (**Bild 9**).

>DRT 10

| LOC | DATA | MNEMONIC | CLIPS | BUS |
|---|---|---|---|---|
| F8BF | FE | OPI | 00000000 | M R F |
| F8C0 | DA | | 00000000 | M R |
| F801 | 08 | RZ | 00000000 | M R F |
| IE90 | BA | | 00000000 | M R |
| IE9D | 30 | | 00000000 | M R |
| 30BA | D3 | OUT | 00000000 | M R F |
| 30BB | F3 | | 00000000 | M R |
| F3F3 | 0A | | 00000000 | I W |
| 30BC | 00 | NOP | 00000000 | M R F |
| 30BD | 76 | HLT | 00000000 | M R F |

>

**Bild 9** Beispiel einer Logik-Analyse

## 6 Schlußbemerkungen

Dieser Artikel kann natürlich nur Anregungen bei der Auswahl eines Mikroprozessor-Entwicklungssystems geben. Der zukünftige Systemanwender wird seine Anforderungen mit den Möglichkeiten der angebotenen Systeme vergleichen und das für ihn optimale Gerät aussuchen.

Zum Kennenlernen eines Entwicklungssystems genügt oft die preiswerte Grundausrüstung, die aber dann nicht mehr ausreicht, wenn das System im tatsächlichen Laboreinsatz ein wertvolles Werkzeug darstellen soll. Das System MDA 8002 A stellt in diesem Sinne ein optimal ausgebautes Entwicklungssystem dar.

**Memory Mapping:** Definierte Speicheraufteilung und Zuweisung.

**Glitch:** Störimpuls oder starker Rauschanteil, der zu Systemfehlern führen kann.

**Triggern:** Takten oder Synchronisieren von Schaltvorgängen (Triggersignal, Triggerimpuls etc.).

Wenn man einen Komplettcomputer anschafft und Erweiterungen sowie den Anschluß zusätzlicher Geräte nicht vorhat, wird man sich um Schnittstellen und Busanordnungen nicht kümmern müssen. Andernfalls kommt man aber an Fachausdrücken wie den folgenden nicht vorbei: V.24-Schnittstelle, RS 232-Spezifikation, 20mA-Stromschleife, Systembus, IEC-Bus etc. Dieser Beitrag gibt einen systematischen Überblick und erklärt diese und weitere Begriffe.

H. Schumny

# Schnittstellen und Busanordnungen

## 1 Schnittstellen

Nach DIN 44302 ist eine Schnittstelle (ein *Interface*) definiert „durch die Gesamtheit der Festlegungen über a) physikalische Eigenschaften der Verbindungsleitungen zwischen zwei Einrichtungen und die auf diesen Leitungen ausgetauschten Signale, b) die Bedeutung der ausgetauschten Signale". Es ist also nicht nur der in **Bild 1** gekennzeichnete Ort damit beschrieben, sondern auch z.B. die Steckerform und -belegung, die Leitungsimpedanzen und die Signalformen sowie deren zeitliche Abläufe.

Die Schnittstellen zwischen Mikrocomputer und Außenwelt (*Peripherie*) können seriell oder parallel ausgeführt sein. „Seriell" nennt man eine Schnittstelle, wenn die in eine Richtung – z.B. vom $\mu$C zur Peripherie – gesendeten Informationen (Daten oder Steuerzeichen) bitseriell, also Bit für Bit nacheinander, auf einer Leitung übertragen werden. „Parallel" arbeitet eine Schnittstelle dann, wenn die Anzahl der Datenleitungen der Zahl der Bit pro Datenwort entspricht, also z.B. 8 oder 16 Bit gleichzeitig (parallel) übertragen werden. In beiden Fällen können die Datenleitungen durch verschiedene Steuerleitungen ergänzt sein, und der Datentransport muß in der Regel in beiden Richtungen möglich sein. Die Tabelle in **Bild 2** gibt Einzelheiten zu den beiden wichtigsten Vertretern serieller Schnittstellen an.

Die *20 mA-Stromschnittstelle* ist nirgendwo offiziell genormt. Sie hat sich trotzdem weltweit durchgesetzt und ist auch als TTY-Schnittstelle bekannt, weil *Teletypes* (TTY) allgemein mit 20 mA-Stromschleifen angeschlossen sind. Besonders hervorzuheben ist, daß sehr große Entfernungen überbrückt werden können. Zusätzliche Leitungen für Steuer- oder Meldesignale sind jedoch nicht vorgesehen.

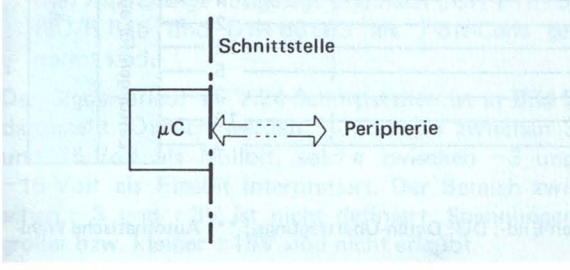



**Bild 1** Schnittstelle zwischen Mikrocomputer ($\mu$C) und Aussenwelt (Peripherie)

| Schnittstelle | genormt nach | Besonderheiten | Anzahl der Leitungen |
|---|---|---|---|
| V.24 | DIN 66020<br>ISO 2110<br>EIA RS 232C<br>CCITT V.24/V.28 | Übertragung mit maximal 20kbit/s über ca. 15–20 m; div. Steuerleitungen | mindestens 3; maximal 25 mit Stecker DSub, 25polig |
| 20mA | – | mehrere Geräte über große Entfernungen anschließbar; keine Steuerleitungen | 4 |

**Bild 2** Serielle Schnittstelle V.24 (Spannungsschnittstelle) und 20mA-Stromschleife

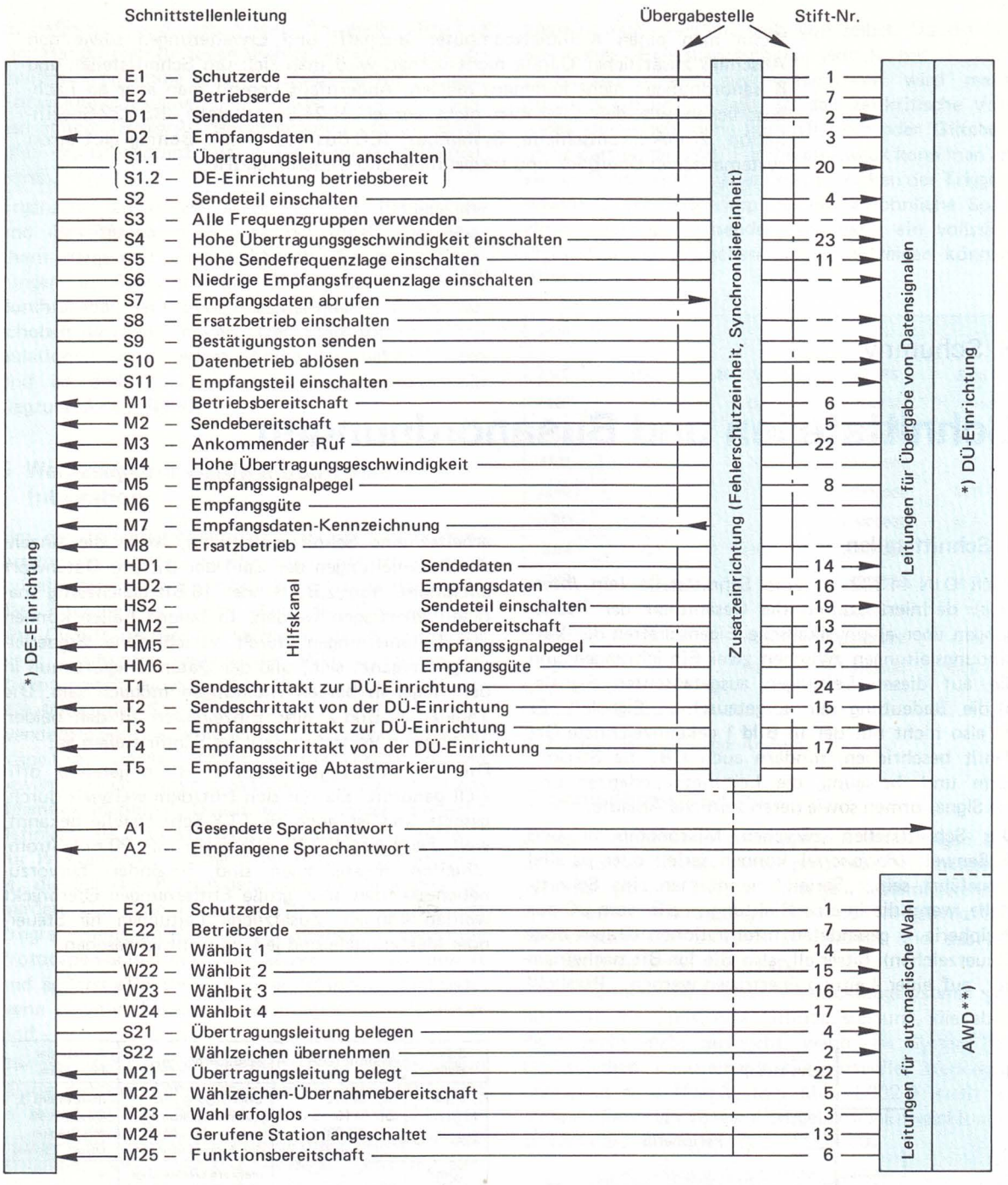



**Bild 3** V.24-Steckerbelegung nach DIN 66020 Blatt 1 (*) DE: Daten-End-; DÜ: Daten-Übertragungs-; **) Automatische Wähleinrichtung für Datenverbindungen)

Die *V.24-Schnittstelle* heißt so, weil sie zuerst vom europäischen CCITT (*Comité Consultatif International Télégraphique et Téléphonique*) als sogenannte Empfehlung V.24 und V.28 veröffentlicht wurde. Von der *International Organization for Standardization* (ISO) wurde diese Empfehlung als ISO-Standard 2110 zur Festlegung der Übergangsstelle (Stiftbelegung der Steckverbinder) herausgegeben. DIN 66 020 Blatt 1 stimmt sachlich mit V.24, V.28 und ISO 2110 überein. In den USA wurde zusätzlich die gleiche Serienschnittstelle von der *Electronic Industry Association* (EIA) als Standard EIA RS 232C genormt. Weil Mikrocomputer ursprünglich aus Amerika kommen, findet man in technischen Angaben eigentlich nie die DIN- oder ISO-Verweise und nur selten die europäische V.24-Bezeichnung, sondern fast immer nur den Hinweis auf RS 232C.

**Bild 3** zeigt die in DIN 66 020 vereinbarte Steckerbelegung. Es scheint nötig, an dieser Stelle ein paar praktische Anmerkungen zu machen:

- Mit den Empfehlungen V.24 und DIN 66 020 wurden alle relevanten Übertragungsfälle abgedeckt, wodurch die enorm große Zahl von Leitungen für Steuer- und Meldezwecke entstand.
- In der Praxis wird jeweils nur eine Auswahl genutzt. Im Extremfall reduziert sich die V.24-Schnittstelle auf drei Leitungen mit
  - *Signalausgang an Pin 2*
  - *Signaleingang an Pin 3*
  - *Masse (Ground) an Pin 7.*

  Das ist auch die Normalkonfiguration von V.24- oder RS 232-Schnittstellen an Mikrocomputern. **Bild 4** zeigt für diesen Fall die genormte Stiftbelegung an der 25-poligen Steckverbindung. Ebenfalls eingezeichnet ist die Schutzerde.
- Die Übertragung der Daten- und Steuerzeichen geschieht mit Hilfe des ASCII-Codes – es werden sogenannte „ASCII-Zeichen" übertragen, Zeichen also, wie sie als *USA Standard Code for Information Interchange* festgelegt und nach CCITT Nr. 5, ISO/R 646 und DIN 66 003 als 7-Bit-Code genormt sind.

Der Signalverlauf an V.24-Schnittstellen ist in **Bild 5** dargestellt. Danach werden Spannungen zwischen 3 und 15 Volt als Nullbit, solche zwischen −3 und −15 Volt als Einsbit interpretiert. Der Bereich zwischen −3 und +3V ist nicht definiert. Spannungen großer bzw. kleiner ± 15V sind nicht erlaubt.

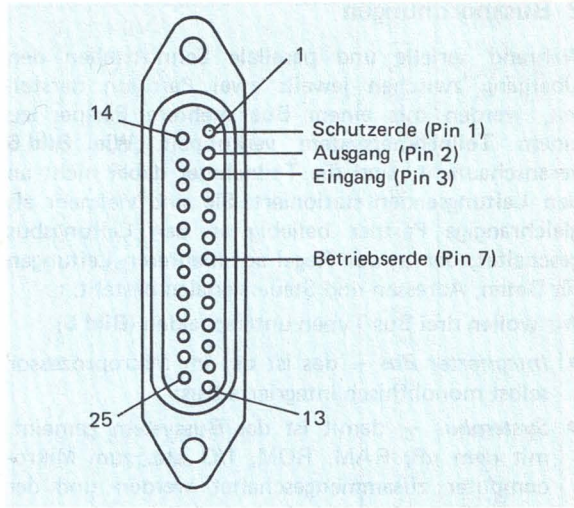



**Bild 4** Genormte 25-polige Steckverbindung der Serie D (DSub, von vorn gesehen) mit Minimalbelegung (Schutzerde wird bei µC-Verbindungen nicht genutzt)

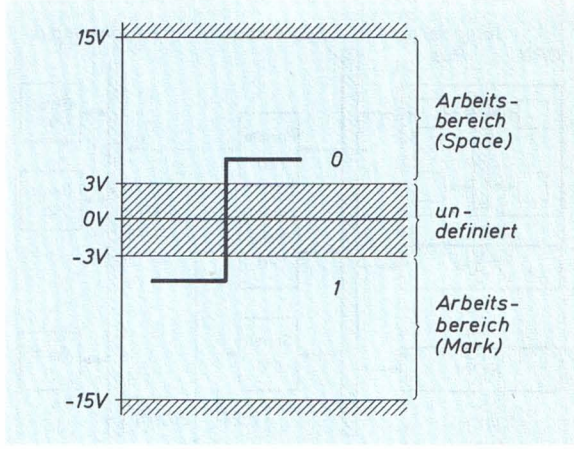



**Bild 5** Signalverlauf an V.24-Schnittstellen

Damit läßt sich zusammenfassen:

- RS 232-Schnittstelle entspricht der V.24-Schnittstelle mit Ausgang an Stift 2, Eingang an Stift 3 und Masse an Stift 7.
- TTY-Schnittstelle schaltet Geräte mit 20 mA-Stromschleifen über vier Leitungen an.

## 2 Busanordnungen

Während serielle und parallele Schnittstellen den Übergang zwischen jeweils zwei Partnern darstellen, werden mit einem Bus mehrere Partner zu einem Teilnehmersystem verkoppelt. Wie **Bild 6** veranschaulicht, sind die Teilnehmer dabei nicht an den Leitungsenden stationiert. Sie sind vielmehr als gleichrangige Partner beliebig an den Leitungsbus geschaltet, der in der Regel aus mehreren Leitungen für Daten, Adressen und Steuersignalen besteht.

Wir wollen drei Bus-Typen unterscheiden (**Bild 6**):

- *Integrierter Bus* – das ist der im Mikroprozessor selbst monolithisch integrierte Bus;
- *Systembus* – damit ist das Bussystem gemeint, mit dem µP, RAM, ROM, I/O etc. zum Mikrocomputer zusammengeschaltet werden und der den weiteren Systemausbau ermöglicht;
- *Peripheriebus* – er erlaubt das gleichzeitige Anschalten mehrerer Peripherie- und Meßgeräte an den µC.

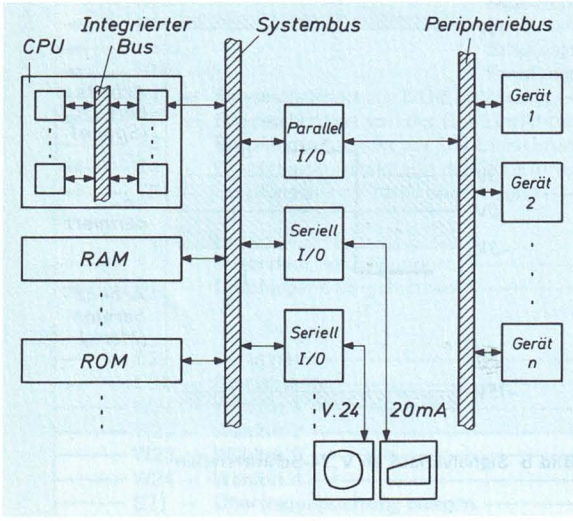



**Bild 6** Mikrocomputersystem mit verschiedenen Bus-Typen und seriellen Schnittstellen

Der in einer CPU (im µP also) integrierte Bus ist für den µP-Benutzer überhaupt nicht „sichtbar". Er wird vom Designer (Entwickler) möglichst optimal ausgeführt, und es besteht gewiß keine Notwendigkeit, solche Busanordnungen zu standardisieren. Anders ist das bei System- und Peripheriebussen. Hierbei ist eine Vereinheitlichung (sprich: Normung) wünschenswert, um Komponenten oder Geräte verschiedener Hersteller anschließen zu können. Im folgenden sollen ein paar wichtige Vertreter von Systembussen und der für Peripherieverwendungen weltweit genormte IEC-Bus vorgestellt werden.

### 2.1 Systembusse

Wenn man sich auf den Standpunkt stellt, daß Mikrocomputer am besten immer aus Komponenten nur eines Herstellers aufgebaut sein sollten, dann erscheint die Frage nach einer herstellerunabhängigen Standardisierung eines Systembusses einigermaßen sinnlos. Dabei wird dann aber übersehen, daß hochspezialisierte Firmen z.B. Speicherkarten oder Ein-Ausgabebausteine viel billiger anbieten als µC-Systemhersteller. Denkbar ist auch der Fall einer Lücke im Lieferprogramm und ein daraus folgender Zwang zum Ausweichen auf einen systemfremden Anbieter.

Solche Überlegungen wurden vermutlich bereits während der „Gründerjahre" des *Micro Computing* angestellt. Denn schon der erste, 1975 vorgestellte *Personal Computer* ALTAIR 8800 von MITS enthielt einen standardisierten Systembus, dessen Spezifikationen von Herstellern und Anwendern gemeinsam festgelegt worden waren. Wegen der inzwischen sehr weiten Verbreitung kann dieser sogenannte S-100 Bus praktisch als wichtigster Standardbus für Mikrocomputer angesehen werden.

*S-100 Bus* steht für „Standardbus mit 100 Leitungen". Die Steckverbindungen sind so ausgeführt, daß zwei 50-polige Stiftreihen parallel liegen. **Bild 7** zeigt die Stiftbelegung. Die folgende Liste erklärt die Bedeutungen.

*Liste der S-100 Bus-Stiftbelegungen:*

| | |
|---|---|
| 1 | 8V für 5V-Spannungsregler |
| 2 | 16V unstabilisiert |
| 3 | External Ready, Warte-Zustand |
| 4–11 | 8 Interrupt-Eingänge (Vectored Interrupt) |
| 12–17 | unbenutzt |
| 18 | Status Disable |
| 19 | Command Control Disable |
| 20 | Reserviert für Speicherschutz |
| 21 | Single Step |
| 22 | Address Disable |
| 23 | Data Out Disable |
| 24 | Clock Phase 2 |
| 25 | Clock Phase 1 |
| 26 | Processor Hold Acknowledge |
| 27 | Processor Wait |
| 28 | Processor Interrupt Enable |
| 29–37 | Adreßeingänge |
| 35–40 | Datenausgänge (Data Out) |

| | |
|---|---|
| 41–43 | Dateneingänge (Data In) |
| 44 | Statusanzeige: Prozessor liest |
| 45 | Statusanzeige: Ausgabe |
| 46 | Statusanzeige: Eingabe |
| 47 | Status für Memory Read |
| 48 | Statusanzeige für HALT-Zustand |
| 49 | Takt |
| 50 | Masse (Ground) |
| 51 | wie 1 |
| 52 | –16V unstabilisiert |
| 53 | Sense Switch Disable |
| 54 | External Clear |
| 55 | Schutzleiter |
| 56–67 | unbenutzt |
| 68 | Memory Write |
| 69 | Reserve für Sonderstatus |
| 70 | Reserve für Speicherschutz |
| 71 | Run/Stop auf RUN |
| 72 | Processor Ready |
| 73 | Processor Interrupt Request |
| 74 | Processor Hold |
| 75 | Processor Reset |
| 76 | Prozessor-Synchronisierung |
| 77 | Processor Write |
| 78 | Processor Data Bus In |
| 79–87 | Adreßeingänge |
| 88–90 | Datenausgänge |
| 91–95 | Dateneingänge |
| 96 | Status Interrupt Accept |
| 97 | Status Write Output |
| 98 | Status Stack Address |
| 99 | Power On Clear |
| 100 | Masse (Ground) |

| Bus | Stife | µC-System | Bemerkungen |
|---|---|---|---|
| ESCO | 96 | ESCO (Neumüller) | Europakartensystem mit 3 X 32-poliger Messerleiste nach DIN 41 612, Bauform C |
| Euro | 64 | Motorola 6800 und µPEP 6408 | Europakartensystem mit Messerleiste nach DIN 41 612, Bauform B 64/64, zweireihig |
| MCS | 64 | MCS ALPHA-1 | Mechanische Ausführung wie beim Euro-Bus |
| MICRO | | National 8080 A | |
| MM | 64 | 8080, Z80, 6502, 6800 | Mechanische Ausführung wie beim Euro-Bus |
| MU | 31<br>44<br>74 | Euromicro | Europakartensystem mit (je nach Anforderung) verschiedenen Steckern |
| Multi | 86 | Intel | zusätzlich 60 „Hilfspins" |
| µCOMP | 56 | PRO-LOG | Platinen 165 X 114 mm |
| STD | 56 | Mosteck | Mechanisch wie µCOMP |
| S 50 | 50 | | |
| S 64 | 64 | PRO COMP (MCDS) | Europakarten mit Stecker nach DIN 41 612, Bauform D 32 |

| | | | |
|---|---|---|---|
| 1 | +8V | 51 | +8V |
| 2 | +16V | 52 | –16V |
| 3 | XRDY | 53 | SSW DSB |
| 4 | VI 0 | 54 | EXT CLR |
| 5 | VI 1 | 55 | * |
| 6 | VI 2 | 56 | |
| 7 | VI 3 | 57 | |
| 8 | VI 4 | 58 | |
| 9 | VI 5 | 59 | |
| 10 | VI 6 | 60 | |
| 11 | VI 7 | 61 | |
| 12 | | 62 | |
| 13 | | 63 | |
| 14 | | 64 | |
| 15 | | 65 | |
| 16 | | 66 | |
| 17 | | 67 | |
| 18 | STATUS DSBL | 68 | MWRITE |
| 19 | CCDSBL | 69 | **** |
| 20 | ** | 70 | *** |
| 21 | SS | 71 | RUN |
| 22 | ADDR DSBL | 72 | PRDY |
| 23 | DO DSBL | 73 | PINT |
| 24 | Ø 2 | 74 | PHOLD |
| 25 | Ø 1 | 75 | PRESET |
| 26 | PHLDA | 76 | PSYNC |
| 27 | PWAIT | 77 | PWR |
| 28 | PINTE | 78 | PDBIN |
| 29 | A 5 | 79 | A 0 |
| 30 | A 4 | 80 | A 1 |
| 31 | A 3 | 81 | A 2 |
| 32 | A 15 | 82 | A 6 |
| 33 | A 12 | 83 | A 7 |
| 34 | A 9 | 84 | A 8 |
| 35 | DO 1 | 85 | A 13 |
| 36 | DO 0 | 86 | A 14 |
| 37 | A 10 | 87 | A 11 |
| 38 | DO 4 | 88 | DO 2 |
| 39 | DO 5 | 89 | DO 3 |
| 40 | DO 6 | 90 | DO 7 |
| 41 | DI 2 | 91 | DI 4 |
| 42 | DI 3 | 92 | DI 5 |
| 43 | DI 7 | 93 | DI 6 |
| 44 | SMI | 94 | DI 1 |
| 45 | SOUT | 95 | DI 0 |
| 46 | SINP | 96 | SINTA |
| 47 | SMEMR | 97 | SWO |
| 48 | SHLTA | 98 | SSTACK |
| 49 | CLOCK (2 MHz) | 99 | POC |
| 50 | GND | 100 | GND |

**Bild 7** Stiftbelegung beim S-100 Bus

Eine Auflistung einiger weiterer Systembusse ist in linksstehender Tabelle vorgenommen. Es handelt sich dabei jedoch um Hersteller- oder Anwender-spezifische Busse, die aber in der Literatur oft genannt werden.

## 2.2 IEC-Bus

Bei der Frage nach dem Sinn der Normung eines Mikrocomputer-Systembusses kann man durchaus geteilter Meinung sein. Kein Zweifel besteht aber daran, daß an den Verbindungsstellen zur Peripherie Standardisierungen notwendig sind. Mit „V.24" und „20mA Current Loop" stehen darum auch zwei Schnittstellen-Verabredungen zur Verfügung, mit deren Hilfe unterschiedlichste Peripheriegeräte an verschiedene $\mu$C angeschlossen werden können. Allerdings ist die Datenübertragung damit nur seriell möglich, und es können nur jeweils zwei Teilnehmer zusammenwirken.

Aus der Meßtechnik entstand der Wunsch, Daten von mehreren Meßgeräten bit-parallel zu einem Rechner übertragen zu können. Gelöst wurde dieses Problem Anfang der siebziger Jahre von der Firma *Hewlett-Packard* durch die Einführung des HP-IB (*Hewlett-Packard-Interface Bus*). Dabei handelt es sich um einen Peripheriebus, der Daten 8bit-parallel überträgt und an den bis zu 15 Geräte angeschlossen werden können. Begleitet sind die Datenleitungen von drei Kontrolleitungen und 5 sogenannten Management-Signalleitungen, der Bus besteht also aus insgesamt 16 Leitungen.

Von den internationalen Normungsgremien wurde der Firmenvorschlag „HP-IB" schnell aufgenommen:

IEEE (*Institute of Electronic and Electrical Engineers*) – Standard IEEE 488/1975;
IEC (*International Electrotechnical Commission*) – Standard IEC 66.22;
DIN (*Deutsches Institut für Normung*) – DIN-IEC 66.22.

Alle drei Normen stimmen sachlich überein und entsprechen dem HP-IB. Leider gibt es aber doch einen Unterschied, der in der Praxis Schwierigkeiten macht:

- Nach DIN-IEC 66.22 ist ein 25-poliger Stecker genormt, der dem V.24-Stecker nach DIN 66 020 entspricht (vgl. **Bild 3**).
- Nach IEEE ist ein 24-poliger Stecker genormt, der dem Hewlett-Packard-Stecker entspricht.

Den Aufbau des IEC-Busses zeigt **Bild 8**. Von den maximal möglichen 15 Teilnehmern sind nur vier eingezeichnet. Anschließbar sind folgende Gerätetypen:

- Geräte, die nur als Sender arbeiten (*Talker*, also „Sprecher");

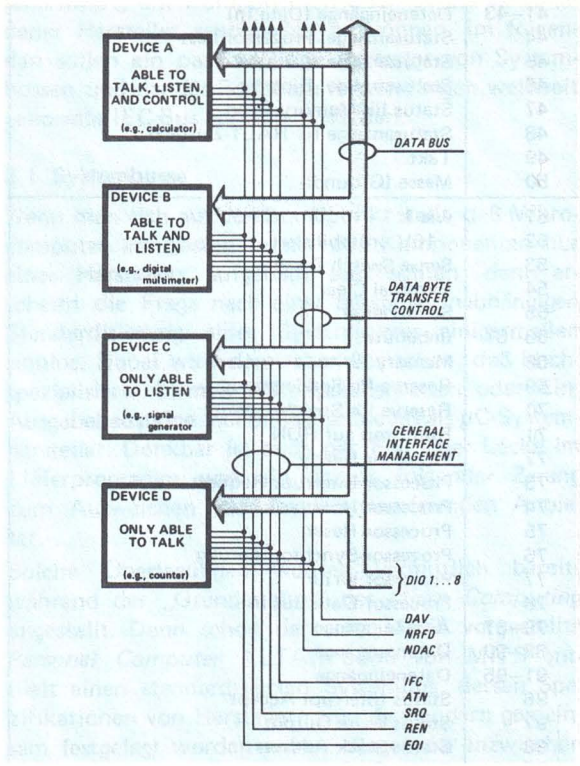



**Bild 8** Strukturschema des IEC-Busses (Abkürzungen und Erklärungen im Text)

- Geräte, die nur als Empfänger arbeiten (*Listener*, also „Hörer");
- Geräte, die „hören" und „sprechen" können;
- ein Gerät, das als *Controller* (Kontrollgerät) wirkt. Dies wird in der Regel ein Rechner sein.

Die acht Kontrolleitungen sind gemäß Bild 8 in zwei Gruppen unterteilt und haben folgende Bedeutungen:

*3 Handshake-Leitungen mit*

DAV – Data Valid (Daten gültig);
NRFD – Not Ready For Data (Nicht aufnahmebereit für Daten);
NDAC – Not Data Accepted (Keine Daten angenommen).

*5 Management-Leitungen mit*

ATN – Attention (Achtung);
IFC – Interface Clear (Interface bereit);
SRQ – Service Request (Service-Anforderung);
REN – Remote Enable (Fernbedienung möglich);
EOI – End Or Identify (Ende oder Kennung).

Abschließend seien ein paar weitere Besonderheiten angeführt:

- Die Datenübertragung ist auf maximal 1Mbyte/s (1 Million Byte pro Sekunde!) festgelegt;
- Die gesamte Kabellänge (Buslänge) darf 20 m nicht überschreiten. D.h. bei Anschaltung aller maximal möglichen 15 Teilnehmer darf die Kabellänge pro Gerät nur knapp 1,50 m betragen.

Der IEC-Bus ist also für schnelle Datenübertragung geeignet. Die Buslänge ist allerdings sehr eingeschränkt. Aus diesem Grunde werden nun Buserweiterungen angeboten, z.B. der Hewlett-Packard *Bus Extender* 37201A. Er wird anstelle eines Teilnehmers an den Bus geschaltet, wandelt die 8bit-parallel ankommenden Daten in einen seriellen Datenstrom um und erlaubt die Überbrückung von bis zu einem Kilometer mit maximal 20kbit/s (synchron) oder 1200 bit/s (asynchron). Mit Hilfe eines *Modems* ist dann auch der Anschluß an das Telefonnetz möglich.

**Zusammenfassung:**

- *Systembusse* verbinden den Mikroprozessor (die CPU) mit Speichern und Ein-Ausgabebausteinen. Sie sind in der Regel Hersteller-spezifisch ausgeführt. Als wichtigster Vertreter einer Hersteller-unabhängigen Normung hat sich der S 100-Bus durchgesetzt.
- *Peripheriebusse* sollen das komplette Computer-System mit Peripheriegeräten (Bedienungsgeräte, Massenspeicher, Meßgeräte etc.) verbinden. Für 8bit-parallele Übertragungen wurde international der IEC-Bus genormt (auch: IEEE 488-Bus).
- *Schnittstellen* für bitserielle Datenübertragungen zwischen Computer und Peripherie sind
  - die V.24-Schnittstelle (auch RS 232-Schnittstelle) nach DIN 66 020 für Bildschirmterminals, Schreibmaschinen etc.,
  - die 20mA-Stromschleife für z.B. Fernschreiber (Teletype, TTY) und große Entfernungen.

## Normen für Datenübertragungseinrichtungen (Modems) und Schnittstellen:

| CCITT | DIN | |
|---|---|---|
| V.19 | | Modems für parallele Datenübertragung mit Telefon-Signalfrequenzen |
| V.20 | | Modems mit Parallelübertragung zur allgemeinen Benutzung im öffentlichen Fernsprechwählnetz (bisher V.30) |
| V.21 | 66021 Teil 1 | Schnittstelle bei 200 bit/s im Gegenbetrieb auf Fernsprechleitungen |
| V.23 | 66021 Teil 2 | Schnittstelle bei 1200 oder 600 bit/s auf Fernsprechleitungen |
| V.25 | 66021 Teil 4 | Schnittstelle bei automatischem Verbindungsaufbau im Fernsprechwählnetzen |
| V.26 | 66021 Teil 3 | Schnittstelle bei 2400 oder 1200 bit/s auf Fernsprechleitungen |
| V.27 | | Schnittstelle bei 4800 bit/s |
| V.29 | | Schnittstelle bei 9600 bit/s |
| V.35 | | Datenübertragung mit 48 kbit/s über Primärgruppenleitungen im Bereich von 60 bis 108 kHz |
| V.36 | | Modems für synchrone Datenübertragung über Primärgruppenleitungen von 60 bis 108 kHz |

Die angegebenen CCITT-Empfehlungen V.19 bis V.36 sind der Empfehlungsserie V entnommen, die für *Datenübertragung über Fernsprechnetze* entwickelt wurde. Für *Datenübertragung über öffentliche Datennetze* gilt die Empfehlungsserie X.

Adressierungsarten, Adreßdecodierung, Adreßraum – das sind Vokabeln aus der „höheren Fachsprache" der fortgeschrittenen Hobby-Computer-Benutzer und der Spezialisten. Nun beinhaltet aber „Hobby Computing" oder „Home Computing", daß auch „Normalverbraucher" etwas von den hinter diesen und weiteren Fachbegriffen verborgenen Zusammenhängen verstehen lernen sollten. Der folgende Beitrag scheint zwar zunächst im Anspruch höher zu liegen. Wegen der klaren und anschaulichen Darstellung werden aber auch „Einsteiger" eine wichtige Portion Grundwissen daraus beziehen können. Es sei darauf hingewiesen, daß in den Bildern die neuen Logik-Schaltzeichen verwendet sind, wie sie nun international und nach DIN 40 700 genormt sind.

P. Welzel

# Die Adressierung in Mikrocomputer-Systemen

## 1 Warum muß der Anwender etwas über Adressierung wissen?

Dem Anwender tritt ein Mikroprozessor meist nicht als ein einzelnes *Chip*, sondern als der Bestandteil eines Systems entgegen. Zusammen mit anderen Bauteilen wie Speichern oder Ein-Ausgabe-Schnittstellen bildet er einen **Minicomputer** oder einen *Mikrocomputer*. Der Benutzer erhält eine Liste, aus der die Adressen der RAM (Lese-Schreibspeicher), der ROM (Festwertspeicher) sowie der **E/A-Schnittstellen** ersichtlich sind, und kann damit arbeiten.

Wer sich aber etwas genauer mit seinem System beschäftigen will, muß sich auch genauer über die *Adressierung* unterrichten, besonders wenn er

- sein System mit weiteren Speicherbausteinen erweitern will;
- neue E/A-Schnittstellen in sein System einfügen will;
- das System so strukturieren will, daß er auch mit Speicherbefehlen Ein-Ausgabe-Vorgänge programmieren kann;
- etwas näher in die **Interrupt-Verarbeitung** einsteigen will, die sich meist ganz bestimmter Speicherplätze bzw. deren Adressen bedient.

## 2 Die Funktion von Adressierung und Adreß-Decodierung

Der Mikroprozessor kennt keine Ein-Ausgabe-Geräte, er kennt keine RAM oder ROM, sondern nur Adressen, auf die er zugreift, d.h. zu lesen oder zu schreiben versucht (in diesem Abschnitt verstehen wir unter Adressen immer die im *Maschinenprogramm* vorhandenen **absoluten Adressen**).

Welche Adressen gibt es für den Mikroprozessor?

a) *Befehlsadressen:* Das Programm ist im Speicher abgelegt (ROM oder RAM), der Befehlszähler (*Program Counter*), der fast immer Bestandteil des Mikroprozessor-Chips ist, enthält die grade benötigte Befehlsadresse.

**Minicomputer:** Bislang noch genutzte Bezeichnung für Mehrzweck-Computer und *Prozeßrechner* mit wenigstens 16 bit Wortbreite und universeller Ausbaufähigkeit. Abgrenzung zum Mikrocomputer wird aber zunehmend schwieriger.

**E/A:** Eingabe/Ausgabe

**Interrupt:** S. „Interrupts in Mikroprozessorsystemen" von *P. Welzel*.

**Absolute Adressen:** Endgültige, nicht mehr änderbare Adressen im Maschinenprogramm.

b) *Operandenadressen:* Dies sind die Adressen, unter denen die zu verarbeitenden Daten abgespeichert sind oder abgespeichert werden sollen. Die Adressen können sowohl RAM-Speicherplätze als auch Ein-Ausgabe-Schnittstellen bezeichnen. Der Prozessor, der diese Adressen auf den **Adreßbus** schaltet, erhält sie aus dem Adreßteil des Befehls oder aus internen Registern.

c) *Stack-Adressen:* der Stack (Stapel) ist ein Bereich des Speichers, der für Sonderaufgaben vorgesehen ist. Er wird über ein internes Register im Prozessor, den *Stackpointer* (Stapelzeiger) adressiert. Zum Zugriff auf die Speicher- und E/A-Bausteine verfügt der Mikroprozessor über seine äußeren Anschlüsse (Pins), die man in einen **Adreßbus, Datenbus, Steuerbus** einteilen kann.

Der Prozessor kann nicht „wissen", welche Bausteine an diese Anschlüsse angeschlossen wurden; er sendet je nach Art des Befehls, der grade abgearbeitet wird, bestimmte Signale zu einer bestimmten Zeit aus und erwartet zu einer bestimmten Zeit Signale, mit denen er weiterarbeiten kann. Hat er z.B. einen Befehl abgeschlossen, so schaltet er den Inhalt des Befehlszählers auf den Adreßbus und gibt ein Steuersignal aus, welches eine Leseoperation im Speicher hervorruft; er erwartet auf dem Datenbus ein *Byte* des neuen Befehls. Damit diese Wirkung eintritt, müssen die Speicherbausteine richtig angeschlossen sein, die auf dem Adreßbus anliegende Adresse muß den gewünschten Speicherplatz anwählen, das vom Prozessor gebildete Steuersignal die Leseoperation auslösen. Dabei ist es oft nicht möglich, alle Leitungen des Adreßbus direkt zu den Speicherbausteinen und E/A-Bausteinen zu führen, es müssen logische Schaltungen durch **Decodierung** für eine gezielte Anwahl sorgen.

## 3 Der Adreßraum

Unter *Adreßraum* versteht man die Anzahl von Speicherplätzen oder E/A-Anschlüssen, die der Prozessor auf Grund seiner *Hardware* und seiner **Befehlsstruktur** ansprechen kann.

Die Zahl der Leitungen des Adreßbus bestimmt den Adreßraum, wobei gilt: $N = 2^n$

$n$ = Zahl der bit am Adreßbus
$N$ = Zahl der Adressen

Nach dem Adreßbus ist meist auch die Befehlsstruktur aufgebaut, d.h. wenn ein Adreßbus mit 16 bit vorliegt, der einen Adreßraum von $2^{16} = 64$ K = 65 536 Adressen erschließt, so muß der Befehl ebenfalls einen Adreßteil von 16 bit umfassen.

Es gibt allerdings Ausnahmen; so haben beim MC 6800, der über einen 16-bit-Adreßbus verfügt und auch über Befehle mit einem 16-bit-Adreßteil, die direkt adressierten Befehle nur einen Adreßteil von 8 bit, der Mikroprozessor kann also mit diesen Speicherbefehlen nur Adressen $\leqslant 255$ ansprechen.

Die Mikroprozessoren unterscheiden sich durch die Art, in der sie die Ein-Ausgabe-Schnittstellen in den Adreßraum einbeziehen; dies geschieht durch eine unterschiedliche Verwendung von Steuersignalen:

### 3.1 Adreßraum bei nicht isolierter Ein-Ausgabe

Bei diesen Systemen (z.B. MC 6800) kann der Prozessor nicht an Hand des Befehles oder der Steuersignale unterscheiden, ob es sich um einen Zugriff zum Speicher oder zur E/A-Schnittstelle handelt; *der Prozessor führt nur die Lese- und Schreiboperationen am Datenbus aus.*

Zur Unterscheidung der beiden Vorgänge genügt grundsätzlich ein Steuersignal (z.B. R/$\overline{W}$), wenn das Signal auf **Low** liegt, wird gelesen, ist es auf **High**, so wird geschrieben. Zur besseren *Synchronisierung*, Unterdrückung von Überlappungen in Umschaltpausen usw. kann ein weiteres Signal verwendet werden, welches die Lese- und Schreibvorgänge auslöst, z.B. VMA: *Valid Memory Address,* d.h. Adresse gültig.

Hinter jeder aufgerufenen Adresse kann sich ein Speicherplatz oder eine E/A-Schnittstelle befinden (siehe **Bild 1**). Der Anwender, der das Programm

**Adreßbus:** Leitungen zur Weiterleitung von Speicheradressen. Meist 16 Leitungen für $2^{16} = 64$ K Speicherstellen (1 K = 1024).

**Datenbus:** Gesamtheit der Datenleitungen; Anzahl entspricht der vorgegebenen Wortbreite (Meist 8 bit = 1 byte, auch 12 oder 16 bit).

**Steuerbus:** Gesamtheit der Leitungen zur Übertragung von Steuersignalen.

**Decodierung:** Entschlüsselung; Erkennen eines Befehls oder Datenwortes und Erzeugen entsprechender Steuersignale.

**Befehlsstruktur:** Aufbau einzelner Befehle und Aufteilung in *Operations- und Adreßteil.*

**Operationsteil** eines Befehls: Gibt an, *was* zu tun ist.

**Adreßteil** eines Befehls: Gibt an, *womit* etwas zu tun ist.

**Low:** Niedriger Signalzustand (z.B. 0 V bei TTL-Schaltungen).

**High:** Hoher Signalzustand (z.B. 5 V bei TTL-Schaltungen).

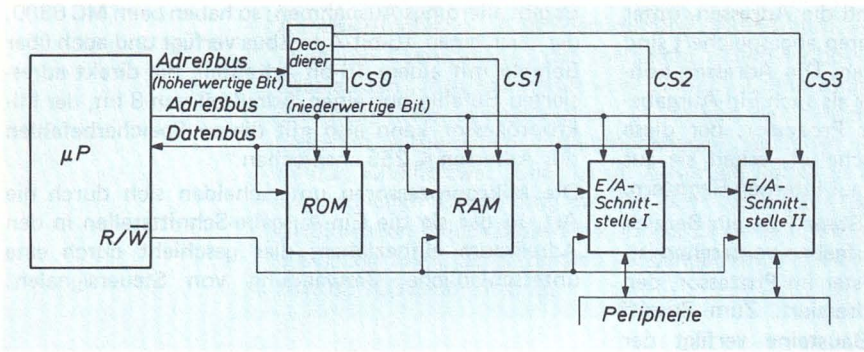



**Bild 1**
Adreßraum bei nicht isolierter E/A
CS: Chip Select (Signal zur Aktivierung des Bausteins)

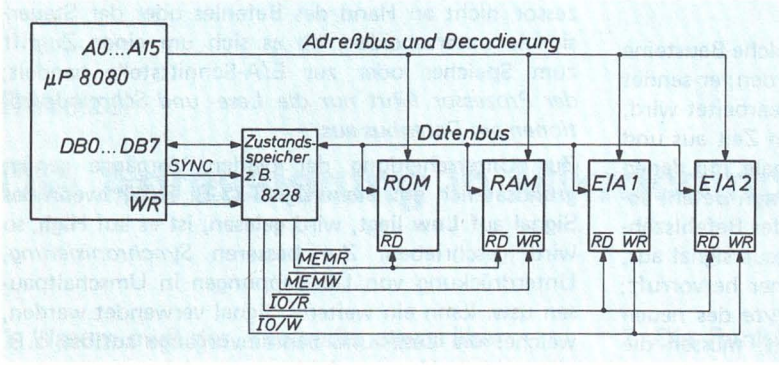



**Bild 2**
System mit isolierter E/A bei Bildung von Einzelsignalen (8080)
Die vier Steuersignale werden von einem eigenen Baustein gebildet auf Grund von Informationen, die zu einer bestimmten Zeit vom Prozessor auf dem Datenbus gebildet werden, sowie des Signals $\overline{WR}$. Die Zustandsinformationen werden durch das Signal SYNC angezeigt; der Baustein dient daneben als Datenbus-Treiber

schreibt, muß natürlich wissen, wozu die Adresse gehört, ob er z.B. ein Abspeichern von Daten oder einen Ausgabevorgang durchführen will. Die beiden Operationen sind aber vom *Operationsteil des Befehls* her nicht zu unterscheiden, sondern nur vom *Adreßteil.*

Da beim Mikroprozessor MC 6800 bestimmte Adressen für die Speicherung der Startadressen von **Interruptprogrammen** bestimmt sind, ist es nicht sinnvoll, unter diesen Adressen E/A-Anschlüsse anzulegen.

### 3.2 Adreßraum bei isolierter Ein-Ausgabe

Bei diesen Systemen wird nicht über die Adresse, sondern durch Steuersignale unterschieden, ob es sich um eine Speicher- oder eine E/A-Operation handelt. Eine Adresse kann einen Speicherplatz *und* einen E/A-Anschluß bezeichnen. Die Steuersignale erzeugt der Prozessor auf Grund der Befehle, die er abarbeitet, bei diesen Systemen gibt es eigene Ein-Ausgabe-Befehle. Zur Bildung der Steuersignale nun **zwei Beispiele:**

a) *Bildung von Einzelsignalen* (siehe **Bild 2**) beim 8080: Die vier möglichen Arbeitsvorgänge „Speicher Lesen", „Speicher schreiben", „Eingabe", „Ausgabe", werden von vier verschiedenen Steuersignalen ausgelöst ($\overline{MEMR}$, $\overline{MEMW}$, $\overline{IO/R}$, $\overline{IO/W}$), der Prozessor sorgt dafür, daß immer nur eins dieser Signale aktiv ist.

b) *Kombination von Signalen* (siehe **Bild 3**) beim 8085: Die vier möglichen Arbeitsvorgänge werden durch eine Kombination von drei Steuersignalen, die der Prozessor direkt bildet (ohne *Zustandsregister*) ausgelöst, der Prozessor sorgt dafür, daß nicht $\overline{RD}$ und $\overline{WR}$ gleichzeitig aktiv sind.

| $\overline{IO/M}$ | $\overline{RD}$ | $\overline{WR}$ | Funktion |
|---|---|---|---|
| 0 | 0 | 1 | Speicher Lesen |
| 0 | 1 | 0 | Speicher Schreiben |
| 1 | 0 | 1 | Eingabe |
| 1 | 1 | 0 | Ausgabe |
| x | 1 | 1 | keine Operation |

**Interruptprogramm:** (*Interrupt Service Routine*). Das Programm, das nach Auftreten eines Interrupts ablaufen soll.

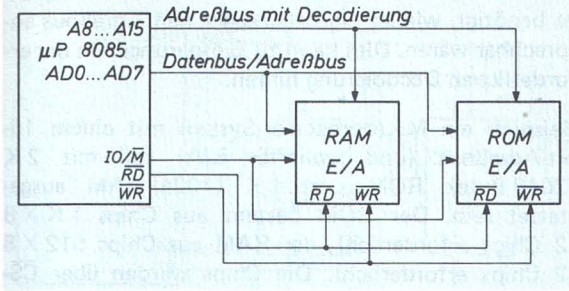



**Bild 3** System mit isolierter E/A bei Bildung von kombinierten Steuersignalen (8085)

Bei den beiden Bausteinen handelt es sich um Chips, die sowohl die Speicher- wie die E/A-Schnittstellenfunktion wahrnehmen.

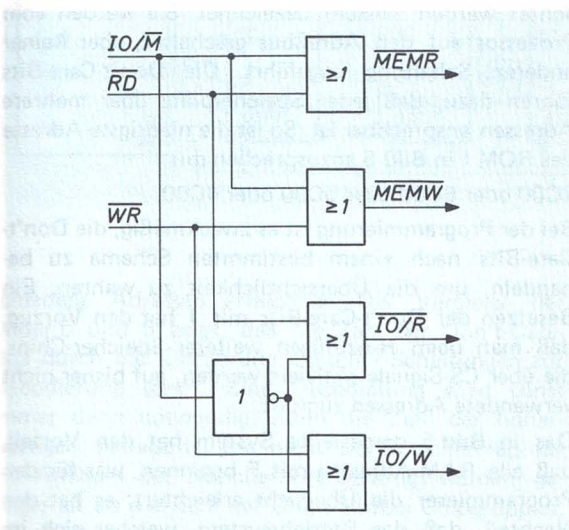



**Bild 4** Umstellung von µP 8080 auf 8085

Bei Umstellung von 8080 auf 8085 kann das Steuersignalverhalten des 8080 durch einfache logische Schaltungen erreicht werden (siehe **Bild 4**). In Systemen mit isolierter E/A ist der Adreßraum bei gleicher Breite des Adreßbus größer als in Systemen mit nicht isolierter E/A, weil alle Adressen mit Speicherplätzen belegt werden können. Da die Zahl der E/A-Anschlüsse geringer ist als die Zahl der Speicherplätze, enthalten die Ein-Ausgabe-Befehle (bei 8080/8085) einen Adreßteil der halben Breite des Adreßbus, es lassen sich $256 = 2^8$ verschiedene E/A-Adressen bilden.

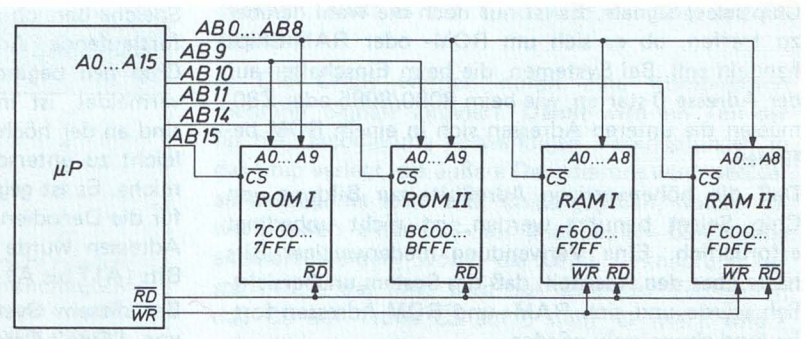
## 4 Aufbau des Speichersystems und Adreß-Decodierung

Der Mikroprozessor bietet auf dem Adreßbus eine Kombination von Bits an, welche einen Speicherplatz bezeichnen. Aufgabe der Decodierung ist es, dafür zu sorgen, daß der durch die Steuersignale bezeichnete Vorgang (Lesen oder Schreiben) genau in diesem Speicherplatz stattfindet. Die Decodierung trifft eine Auswahl „1 aus *N*" (*N*: Anzahl aller Adressen). Für den Anwender gliedert sich die Decodierung in zwei Ebenen, die sich von der Logik her aber in keiner Weise unterscheiden:

### 4.1 Voraussetzungen für die Decodierung vom Prozessor

Beim Aufbau des Systems wird der Speicher aus einzelnen Chips gebildet. Die Aufgabe der Decodierung ist es, davon einen oder mehrere Chips auszuwählen (die Auswahl mehrerer Chips zu einer Zeit erfolgt, wenn die Datenwortbreite des Speichers kleiner ist als die Breite des Datenbus). Die Auswahl der Chips erfolgt auf Grund der höherwertigen Adreßbits durch logische Schaltungen. Diese bilden die *Chip-Select-Signale*, welche die einzelnen Chips aktivieren (siehe **Bild 5**).

**Bild 5**
Detaillierte Darstellung der Chip-Select-Signale (vgl. Bild 1)
AB: Adreßbus



**Beispiel:** Ein System mit einem 16-bit-Adreßbus bei isolierter E/A verfügt über Speicher-Chips von 4 K X 8 bit (RAM oder ROM). Das Speichersystem muß aus 16 Chips ausgeführt werden, damit werden die 4 höchstwertigen Bits des Adreßbus zur Bildung der Chip-Select-Signale herangezogen (Auswahl 1 aus 16), die niederwertigen 12 Bits dienen der Auswahl der Speicherzelle innerhalb des Chips.

### 4.2 Voraussetzungen für die Decodierung vom Speicher-Chip

Jedes Speicher-Chip enthält mehrere Speicherplätze (*Locations*). Eine im Chip vorhandene Decodierung sorgt für die Auswahl eines Speicherplatzes auf Grund der Adreßeingänge des Chips. Die Zahl der Adreßeingänge des Chips richtet sich grundsätzlich nach der Beziehung

$A = 2^b$

$A$ = Anzahl der Speicherplätze im Chip
$b$ = Adreßanschlüsse am Chip

Die Adreßeingänge der Speicher-Chips werden direkt an den niederwertigen Adreßbits des Adreßbus angeschlossen (siehe **Bild 5**).

Für den Aufbau des Speichersystems und seiner Decodierung spielt es keine Rolle, ob es sich um ROM- oder RAM-Chips handelt. Für die Interrupt-Verarbeitung kann es aber wichtig sein, daß bestimmte Adressen sich in ROM-Plätzen befinden, da ein ROM auch bei Netzausfall oder ähnlichen Störungen nicht die Information verliert.

### 4.3 Beispiele für System-Konfigurationen, Don't-Care-Bits

Wenn der Adreßraum eines Mikroprozessors voll ausgenutzt wird, d. h. zu jeder möglichen Adresse tatsächlich ein Speicherplatz gehört, bestehen für den Aufbau des Systems keine Wahlmöglichkeiten, die niederwertigen Adreßbits werden direkt den Chips zugeführt, die höherwertigen dienen der Bildung der Chip-select-Signale. Es ist nur noch die Wahl darüber zu treffen, ob es sich um ROM- oder RAM-Chips handeln soll. Bei Systemen, die beim Einschalten auf der Adresse 0 starten, wie beim 8080/8085 oder Z80, müssen die unteren Adressen sich in einem ROM befinden.

Daß die höherwertigen Adreßbits zur Bildung von Chip Select benutzt werden, ist nicht unbedingt erforderlich. Eine Verwendung niederwertiger Bits hätte aber den Nachteil, daß das System unübersichtlich würde und sich RAM- und ROM-Adressen fortlaufend abwechseln würden.

In vielen Systemen werden nicht soviele Speicherplätze benötigt, wie sie eigentlich über den Adreßbus ansprechbar wären. Dies kann zu Einsparungen in der erforderlichen Decodierung führen.

**Beispiel:** ein Mikroprozessor-System mit einem 16-bit-Adreßbus (und isolierter E/A) soll mit 2 K (2048 Byte) ROM und 1 K (1024) RAM ausgestattet sein. Der ROM besteht aus Chips 1 K X 8 (2 Chips erforderlich), der RAM aus Chips 512 X 8 (2 Chips erforderlich). Die Chips werden über $\overline{CS}$-Signale aktiviert (*Low* = aktiv).

Das geforderte System kann in verschiedenen Variationen aufgebaut werden. Bei der in **Bild 5** gezeigten Adreßaufteilung ist angenommen, daß alle Bits, die nicht zur Ansteuerung des Speichers benötigt werden, mit 1 geschrieben wurden. Diese Bits (A12 und A13) werden als *Don't-Care-Bits* (etwa: Bits, die nicht beachtet werden müssen) bezeichnet. Sie werden vom Prozessor auf den Adreßbus geschaltet, aber keiner anderen Schaltung zugeführt. Die Don't-Care-Bits führen dazu, daß jeder Speicherplatz über mehrere Adressen ansprechbar ist. So ist die niedrigste Adresse des ROM I in **Bild 5** anzusprechen mit

7C00 oder 6C00 oder 5C00 oder 4C00.

Bei der Programmierung ist es zweckmäßig, die Don't-Care-Bits nach einem bestimmten Schema zu behandeln, um die Übersichtlichkeit zu wahren. Ein Besetzen der Don't-Care-Bits mit 1 hat den Vorzug, daß man beim Hinzufügen weiterer Speicher-Chips, die über $\overline{CS}$-Signale aktiviert werden, auf bisher nicht verwendete Adressen zugreift.

Das in **Bild 5** dargestellte System hat den Vorteil, daß alle RAM-Adressen mit F beginnen, was für den Programmierer die Übersicht erleichtert; es hat den Nachteil, daß das Betriebssystem, welches sich im ROM befindet, nicht auf der Adresse 0 beginnt, was bei einigen Prozessoren wegen der Interruptverarbeitung und dem Rücksetzverhalten notwendig ist. Ein weiterer Nachteil besteht darin, daß der RAM-Speicherbereich und der ROM-Speicherbereich keine fortlaufende Adressierung erhalten, sondern jedes Chip neu beginnt. Ein Aufbau, der diese Nachteile vermeidet, ist in **Bild 6** zu sehen. RAM und ROM sind an der höchsten Hexadezimal-Ziffer der Adresse leicht zu unterscheiden und bilden fortlaufende Bereiche. Es ist gegenüber **Bild 5** ein erhöhter Aufwand für die Decodierung notwendig. Bei den angegebenen Adressen wurde angenommen, daß alle Don't-Care-Bits (A11 bis A14) auf 0 gesetzt wurden.

Bei diesem System ist wegen des Vorhandenseins von 4 Don't-Care-Bits jede Speicherzelle über 16 ver-

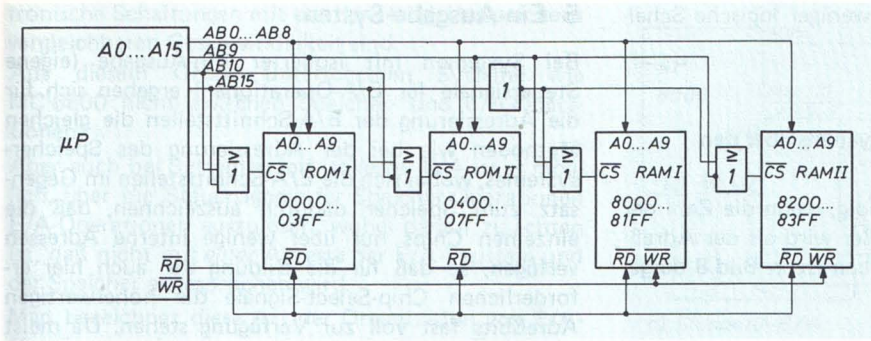



**Bild 6**
Speicherorganisation mit fortlaufender Adressierung

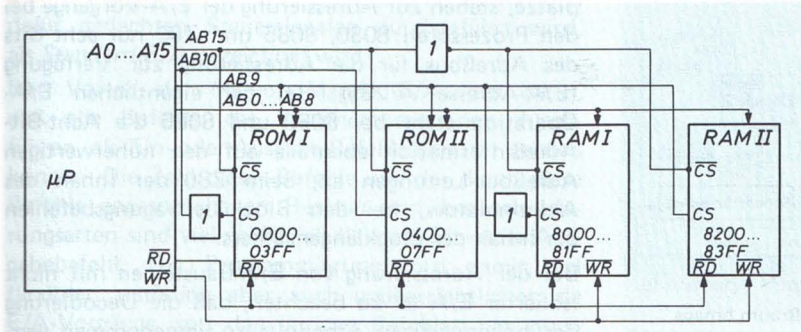



**Bild 7**
Chip-Aktivierung durch Kombination mehrerer Signale

schiedene Adressen erreichbar. Der Vergleich der Bilder 5 und 6 zeigt, daß die Zahl der Don't-Care-Bits durch die Einführung logischer Schaltungen zur Decodierung steigt. Eine Decodierung wird sonst immer dann notwendig, wenn die Zahl der höherwertigen Adreßbits, die nicht am Speicherchip für die Auswahl der Speicherzelle benötigt werden, geringer ist als die Zahl der Chips (wobei Chipgruppen, die gleichzeitig aktiviert werden, als 1 Chip zählen).

Bei einem 64-K-System, welches mit Speicherchips von 1 K aufgebaut wird, gilt:

| Größe des vorhandenen Speichers (in K = 1024) | Zahl der Don't-Care-Bits mit Decodierung | ohne Decodierung |
|---|---|---|
| 1 | 6 | 6 |
| 2 | 5 | 4 |
| 3 | 4 | 3 |
| 4 | 4 | 2 |
| 5 | 3 | 1 |
| 6 | 3 | 0 |
| 7– 8 | 3 | nicht möglich |
| 9–16 | 2 | nicht möglich |
| 16–32 | 1 | nicht möglich |
| 32–64 | 0 | nicht möglich |

Wird das System nur mit einem Speicherchip ausgestattet, so kann das Chip-Select-Signal konstant an 0 V gelegt werden. Dieser Fall dürfte in der Praxis aber nie auftreten.

Die Decodierung läßt sich vereinfachen, wenn nicht alle Speicherbausteine durch einen $\overline{CS}$-Eingang mit dem gleichen aktiven Signal angesprochen werden:

a) Für RAM und ROM gelten unterschiedlichen Signale als aktiv (z. B. CS bei RAM, $\overline{CS}$ bei ROM), wie es z. B. im 8085-System vorkommt. Mit einem Adreßbit lassen sich dann zwei Chips unterscheiden. Steht dieses Bit auf 0, so wird das Chip mit $\overline{CS}$ aktiviert, steht es auf 1, so wird das Chip mit CS aktiviert.

b) Die Chips werden durch eine Kombination mehrerer Signale aktiviert. Damit wird ein Teil der für die Decodierung notwendigen Verknüpfungen in das Chip verlegt, die äußere Decodierung wird weniger aufwendig. Es seien z. B. Chips gegeben, welche sowohl durch einen CS- als auch einen $\overline{CS}$-Eingang aktiviert werden, wobei die UND-Verknüpfung besteht, d. h. das Chip wird nur unter der Bedingung, daß CS auf 1 und $\overline{CS}$ auf 0 steht, aktiviert. **Bild 7** zeigt für solche Chips die Ansteuerung. Der Vergleich

mit **Bild 6** zeigt, daß hierbei weniger logische Schaltungen benötigt werden.

### 4.4 Erweiterung des Speichersystems über den Adreßraum hinaus

Die Erweiterung wird notwendig, wenn die Zahl der benötigten Speicherplätze größer wird als der Adreßraum. Der grundsätzliche Aufbau ist in **Bild 8** dargestellt.

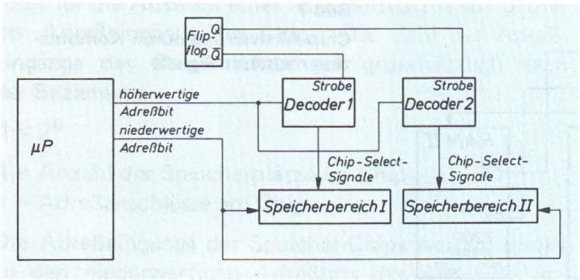



**Bild 8** Speichererweiterung über den Adreßraum hinaus

Das **Flipflop** sorgt dafür, daß nur einer der Decodierer das **Strobe-Signal** erhält; der Decodierer, der nicht das aktive Strobe-Signal erhält, schaltet alle seine Ausgänge auf 1, der Decodierer mit dem aktiven Strobe-Signal bildet ein aktives CS-Signal (*Low*).

Das Flipflop wird gesetzt und gelöscht durch Ausgabe-Befehle. Unter der Voraussetzung, daß das Flipflop die Ausgabe-Kanaladresse F8 hat, müßte bei Erreichen der oberen Grenze des Speichers I programmiert werden:

```
I  FFFC   MVI  A,01H
I  FFFE   OUT  F8H
II 0000   Erster Befehl des neuen Speicher-
          bereichs
```

Das Programmbeispiel ist der **Assemblersprache** des 8080/8085/Z80 entnommen. Der Ausgabebefehl setzt mit der in den *Akku* übernommen 1 das Flipflop, der Befehlszähler des Prozessors steht nach Abarbeitung des Ausgabebefehls auf 0000. Gelesen wird der Befehl aus dem Speicherbereich II.

## 5 Ein-Ausgabe-System

Bei Systemen mit isolierter Ein-Ausgabe (eigene Steuersignale für E/A-Operationen) ergeben sich für die Adressierung der E/A-Schnittstellen die gleichen Methoden wie bei der Adressierung des Speichersystemes, wobei sich die E/A-Schnittstellen im Gegensatz zum Speicher dadurch auszeichnen, daß die einzelnen Chips nur über wenige interne Adressen verfügen, so daß für die Bildung der auch hier erforderlichen Chip-Select-Signale die höherwertigen Adreßbits fast voll zur Verfügung stehen. Da meist weniger E/A-Anschlüsse benötigt werden als Speicherplätze, stehen zur Adressierung der E/A-Vorgänge bei den Prozessoren 8080, 8085 und Z80 nur acht Bits des Adreßbus für die Adressierung zur Verfügung (E/A-Adresse $\leqslant 255$). Bei der eigentlichen E/A-Operation steht bei 8080 und 8085 die Acht-Bit-Adreßinformation ebenfalls auf den höherwertigen Adreßbus-Leitungen an, beim Z80 der Inhalt des Akkumulators, bei den Blockübertragungsbefehlen der Inhalt des Blocklängenzählers.

Bei der Adressierung von E/A-Bausteinen mit nicht isolierter E/A ist zu beachten, daß die Decodierung der höherwertigen Adreßbits so vorgenommen werden muß, daß nicht gleichzeitig E/A- und Speicherbausteine angesprochen werden, da diese ja über die gleichen Steuersignale adressiert werden.

### 5.1 Memory-Mapped-E/A

In einem Mikroprozessor-System unterscheidet sich ein Eingabe-Befehl nicht grundsätzlich von einer Speicher-Lese-Operation, es handelt sich um einen Datentransport von außen in den Prozessor. Es unterscheidet sich ein Ausgabebefehl nicht grundsätzlich von einem Speicher-Schreib-Befehl, es handelt sich um einen Datentransport vom Prozessor nach außen.

Auch vom Zeitverhalten her besteht kein Unterschied. Zwar sind die peripheren Geräte meist wesentlich langsamer als der Mikroprozessor, aber dieser verkehrt nicht mit den peripheren Geräten, sondern mit den Schnittstellenbausteinen, welche rein elek-

**Flipflop:** Grundschaltung für Halbleiterspeicher (bistabile Kippstufe).

**Strobe-Signal:** Steuersignal das aktiv ist, wenn die auf den Bus gelegten Daten korrekt sind.

**Decoder:** Decodierer.

**Assemblersprache:** Prozessor-spezifische Programmiersprache, bei der jeder mnemonische Ausdruck einem Maschinenbefehl entspricht.

tronische Schaltungen mit der den Halbleiterspeichern vergleichbaren Geschwindigkeit sind.

Aus diesem Grund unterscheiden Systeme wie MC 6800 nicht zwischen Speicher- und E/A-Operationen.

Aber auch bei Systemen mit isolierter E/A ist es möglich, über die Steuersignale für Speicher-Operationen E/A-Operationen auszulösen, wobei darauf zu achten ist, daß nicht mit *einer* Adresse der E/A-Baustein und der Speicher angesprochen wird.

Man bezeichnet diese Art der Organisation von E/A-Vorgängen als die *speicher-orientierte E/A* oder **Memory-Mapped-E/A**, während die E/A, die mit den dafür gedachten Steuersignalen durchgeführt wird, als *Standard-E/A* bezeichnet wird.

Der Vorteil der Memory-Mapped-E/A liegt darin, daß alle Befehle, die Speicheroperationen durchführen, als Ein- oder Ausgabe-Befehle benutzt werden können. Die Zahl dieser Befehle und der durch diese Befehle angesprochenen Register und ihre Adressierungsarten sind viel größer als bei dem Ein- oder Ausgabebefehl. Die Programmierung wird damit viel flexibler, natürlich aber auch unübersichtlicher, da E/A-Vorgänge von den übrigen Befehlen nur noch über die Adresse unterscheibar sind.

Die Verwendung von Standard-E/A und Memory-Mapped-E/A soll am Beispiel einer Minimalkonfiguration des 8085-Systems dargestellt werden (**Bild 9**). Es werden hier Bausteine verwandt, die sowohl einen Speicher (RAM oder ROM) wie auch E/A-Schnittstellen enthalten (außerdem programmierbare Zeitgeber). Zur Entscheidung darüber, ob der E/A-Anschluß oder der Speicher aktiviert werden soll, ist das Steuersignal IO/$\overline{M}$ des Prozessors gedacht, wofür an den externen Bausteinen ein Eingang vorhanden ist. Bei **Bild 9b** wird aber mit Adreßbit 15 über Speicher oder E/A entschieden, Adreßbit 14 entscheidet über die Baustein-Anwahl. Da für Lesen das Signal $\overline{RD}$, für Schreiben das Signal $\overline{WR}$ gebildet wird, das Signal IO/$\overline{M}$ nicht ausgewertet wird, können alle E/A- und Speicheroperationen mit allen E/A- und Speicherbefehlen ausgeführt werden.

## 6 Zeit-Multiplex-Verfahren bei der Adressierung

Nicht immer steht für jede Information, die für den Verkehr zwischen Mikroprozessor und seinen externen Komponenten benötigt wird, eine Leitung dauernd zur Verfügung, sondern es muß in einigen Fällen eine Leitung zu verschiedenen Zeitpunkten verschieden genutzt werden. Während es Halbleiterspeicher gibt,

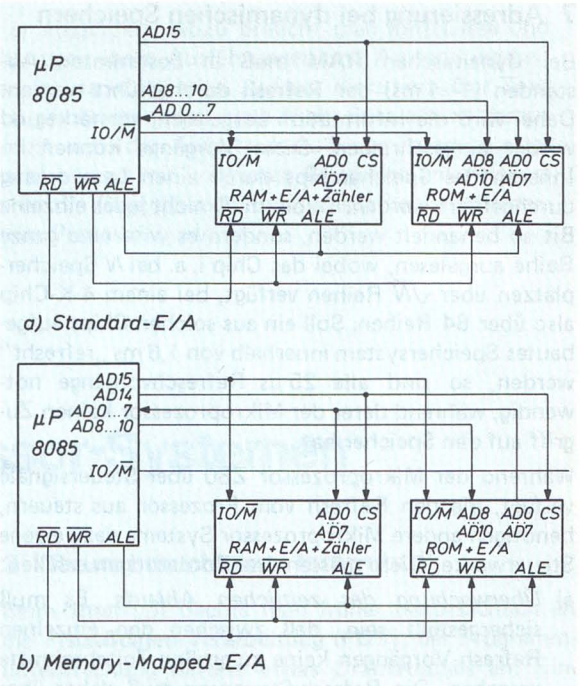



**Bild 9** Ein-Ausgabemethoden

welche die Adresse als Zeilen- und Spaltenadressen zeitlich gestaffelt zugeführt erhalten, kann bei einem Mikroprozessor-System eine Verbindung von Datenbus und Adreßbus stattfinden, wie es beim 8085 der Fall ist.

Wie in **Bild 9** zu sehen ist, werden die niederwertigen 8 Bit des Adreßbus auch als Datenbus genutzt, während die höherwertigen 8 bit nur als Adreßbus zur Verfügung stehen. Das Steuersignal ALE zeigt an, daß auf den Leitungen AD 0 . . . 7 Adressen anstehen. Mit der negativen Flanke von ALE (**Address Latch Enable**) müssen die Adressen in einen Zwischenspeicher übernommen werden. Dieser kann sich direkt im Speicher-EA-Chip befinden, wie in **Bild 9** dargestellt, er kann auch extern angebracht sein.

Dieses Verhalten bringt für den Mikroprozessor keine zeitliche Verzögerung. Es wird in T1 des Operationszyklus die Adresse gebildet, während der Datenverkehr erst während T3 des Zyklus stattfindet.

**Memory-Mapped-E/A:** Ein-Ausgabe mit festgelegter Speicherbereichszuweisung.

**Address Latch Enable:** Ermöglichen (*Enable*) der Übernahme einer Adresse in einen Zwischenspeicher (*Latch*).

## 7 Adressierung bei dynamischen Speichern

Bei dynamischen RAM muß in bestimmten Abständen (1–4 ms) der **Refresh** durchgeführt werden. Dabei wird die Information ausgelesen, verstärkt und wieder eingeschrieben. Diese Vorgänge können im Inneren des Speicherchips durch einen Lesevorgang durchgeführt werden. Dabei muß nicht jedes einzelne Bit so behandelt werden, sondern es wird eine ganze Reihe ausgelesen, wobei das Chip i. a. bei $N$ Speicherplätzen über $\sqrt{N}$ Reihen verfügt, bei einem 4-K-Chip also über 64 Reihen. Soll ein aus solchen Chips aufgebautes Speichersystem innerhalb von 1,6 ms „refresht" werden, so sind alle 25 $\mu$s Refreschvorgänge notwendig, während derer der Mikroprozessor keinen Zugriff auf den Speicher hat.

Während der Mikroprozessor Z80 über Steuersignale verfügt, die den Refresh vom Prozessor aus steuern, benötigen andere Mikroprozessor-Systeme dazu eigene Steuerwerke. Diese müssen drei Forderungen erfüllen:

a) *Überwachung des zeitlichen Ablaufs.* Es muß sichergestellt sein, daß zwischen den einzelnen Refresh-Vorgängen keine zu großen Zeitabschnitte vergehen. Die Refresh-Steuerung muß daher über eine Zeitschaltung verfügen, diese kann aus dem Prozessor-Takt oder aus einem eigenen Takt gespeist werden.

b) *Überwachung der Reihenadressen.* Die Steuerung muß über einen Adreßzähler für die Reihenadressen verfügen, diese müssen bei Durchführung des Refreshvorgangs auf die Adreßeingänge der Chips geschaltet werden, wenn sie direkt am Datenbus des Prozessors liegen. So verfügt der Prozessor über Möglichkeiten, von sich aus den Adreßbus logisch neutral zu halten, so daß mit jeder **Open-Collector-** bzw. **Three-State-Schaltung** die Adressen aufzuschalten sind. Andernfalls können die Adressen über eine Multiplexerschaltung zugeführt werden.

c) *Beeinflussung des Prozessors.* Da der Prozessor während des Refresh-Vorgangs nicht auf den Speicher zugreifen kann, muß dafür gesorgt sein, daß er in seiner Arbeit in den Wartezustand geht und den Adreßbus logisch neutral schaltet. Dabei muß die Arbeit der Refresh-Steuerung so auf den Prozessor abgestimmt sein, daß er nicht laufenden Datentransporte behindert, sondern erst nach Abschluß eines Datentransports (Operationszyklus bei 8080, 8085 oder Z80) wirksam werden kann. Die Refresh-Steuerung muß den Prozessor dazu über die dafür vorgesehenen Steuersignale beeinflussen, z. B. über HOLD.

### Literatur

*Barden:* The Z80 microcomputer handbook. 1978.

*Motorola:* M 6800 Application Manual. 1975.

*Intel:* Data Catalog. 1077.

*Siemens:* Mikroprozessor-Bausteine System SAB 8080.

*Titus, Titus, Larsen, Rony:* Micro-Data Stack/Processors and Elements. Computer Design 1977–1979 (Serie).

*Dollhoff:* Micro-Prozessor Software: How to Optimize Timing & Memory Usage. Digital Design Febraur 1977.

*Motorola:* M 6800 Microprocessor Programming Manual. 1975.

Fifth Annual Microprocessor Directory. EDN 21/1978.

**Refresh:** Auffrischung, Wiederherstellung von Speicherinhalten.

**Open Collector:** (offener Kollektor). Verfahren zur Zusammenschaltung mehrerer Ausgangsstufen von digitalen integrierten Schaltungen.

**Three-State-Schaltung:** Ausgangsschaltung, die zusätzlich zu *High* und *Low* in einen dritten, hochohmigen Zustand versetzt werden kann und damit keine Belastung für vorhergehende Stufen mehr darstellt (auch: *Tristate*).

Was ist ein Interrupt, wie wird er ausgeführt, wozu braucht man ihn? Diese und viele weitere Fragen, die mit Interrupts, ihren Ausführungen und Anwendungsbeispielen zusammenhängen, werden im nachfolgenden Aufsatz diskutiert. Der Text ist so ausführlich und verständlich geschrieben, daß auch Nicht-Spezialisten daraus etwas lernen können. Für Fachleute wird der Aufsatz deshalb von Interesse sein, weil die Diskussion sehr umfassend ist und mit konkreten Beispielen angereichert wurde. Besonders hervorzuheben ist die Gegenüberstellung unterschiedlicher Interrupt-Konzeptionen an verschiedenen Prozessortypen.

P. Welzel

# Interrupts in Mikroprozessor-Systemen

## 1 Was ist ein Interrupt?

Ein Interrupt ist das Unterbrechen eines laufenden und der Start eines anderen Programms. Der Zeitpunkt, an dem das laufende Programm unterbrochen wird, ist durch ein von außerhalb des Prozessors kommendes Signal (**Interrupt Request**) bestimmt, welches zu jedem beliebigen Zeitpunkt, also unvorhersehbar, eintreffen kann. Grundsätzlich wird das durch einen Interrupt unterbrochene Programm fortgesetzt, nachdem das dadurch gestartete Programm (**Interrupt Service Routine**) abgearbeitet worden ist.

## 2 Wie wird ein Interrupt ausgeführt?

- Das Anforderungs-Signal für den Interrupt kommt an.
- Der Prozessor muß verschiedene Bedingungen abprüfen, um zu entscheiden, ob er das laufende Programm unterbrechen darf.
- Wenn die Entscheidung positiv ausfällt, müssen Informationen über den Zustand des Prozessors sichergestellt werden.
- Danach oder gleichzeitig damit muß der Prozessor feststellen, welches neue Programm gestartet werden soll, was sich nach der Ursache des Interrupts richtet.
- Die Startadresse der Interrupt Service Routine wird in den Befehlszähler übernommen, der nächste Befehl (erster Befehl der Interrupt Service Routine) wird abgearbeitet.

## 3 Warum braucht man Interrupts?

Beim Interrupt dachte man früher hauptsächlich an die **Prozeßdaten-Verarbeitung** (PDV), der Programmunterbrechung haftete etwas Dramatisches an. Eingänge, von denen das Interrupt-Request-Signal dem Prozessor zugeführt wurden, nannte man *„Alarmeingänge"*.

Heute gehört die Interrupt-Abarbeitung zur normalen Arbeitsweise einer informationsverarbeitenden Anlage; er dient der Kommunikation mit den Komponenten des Systems. Einige Beispiele sind:

a) *Zeit-Interrupt:* Ein Taktgeber unterbricht in regelmäßigen Abständen das Programm, z. B. alle 100 ms; der Prozessor kann auf Grund dieser Unterbrechungen eine Tageszeituhr mitzählen, Aufträge bestimmter Dauer ausführen, in richtigen zeitlichen Abständen Meßstellen abfragen, seine Betriebsmittel auf verschiedene Aufträge zeitlich aufteilen (**Time Sharing**).

**Interrupt Request:** (IRQ). Anfrage an CPU, ob Interrupt ausgeführt werden kann. Wird sofort nach Abschluß der gerade laufenden Instruktion akzeptiert, wenn nicht wichtigeres dagegen spricht.

**Interrupt Service Routine:** Interrupt-Dienstprogramm. Wird ausgeführt, wenn IRQ akzeptiert wurde.

**Prozeßdatenverarbeitung:** (PDV). Moderner Zweig der EDV, der sich mit der Steuerung von industriellen oder wissenschaftlichen Prozessen und der Weiterverarbeitung der daraus kommenden Meßwerte befaßt.

**Time Sharing:** (*Teilnehmerbetrieb* oder *Zeitscheiben-Technik*). Bei diesem Verfahren haben viele Computer-Benutzer praktisch gleichzeitig Zugriff zu ihrem eigenen Programm.

b) *Ein-Ausgabe-Interrupt:* Kann auftreten, wenn Daten eingegeben werden sollen, z. B. die Tastatur betätigt wurde, oder Daten ausgegeben werden sollen, z. B. ein Drucker die Zeile beendet hat und neue Zeichen benötigt.

c) **DMA-Interrupt:** Für schnelle Daten-Ein- und Ausgaben wird der E/A-Verkehr oft durch einen *DMA-Controller* (Direktspeicherzugriffs-Steuerung) gesteuert. Der Prozessor wird durch Interrupts vom Ende solcher DMA-Vorgänge unterrichtet (*Blockende-Interrupt*), aber auch bei Störungen im Datenverkehr. Auf diese Weise kann der DMA-Controller selbsttätig arbeiten, und der Prozessor kann einen Überblick über den Stand der Arbeiten behalten.

d) **Analog-Digital-Wandler** benötigen für eine Umsetzung oft einen erheblichen Zeitraum, bei integrierenden Wandlern oft über 20 ms. Der Prozessor stößt den Umsetzungsvorgang an, der AD-Wandler bildet nach Abschluß einen Interrupt, damit der Prozessor die Daten einlesen kann.

e) *Maschinenfehlerinterrupts, Netzausfallinterrupt, Paritätsverletzungsinterrupt,* dazu gehören auch die Interrupts, die erzeugt werden, wenn sich ein Gerät oder eine Komponente nicht innerhalb einer bestimmten Zeit auf ein Signal hin gemeldet haben (*Time Out Interrupt*).

f) *Alarm- und Fehlmeldungen* aus einem zu überwachenden Prozeß oder einer zu steuernden Anlage.

## 4 Wodurch unterscheidet sich der Interrupt von einem Unterprogrammaufruf?

*Interrupt* und *Unterprogrammaufruf* haben viele Gemeinsamkeiten; beide führen zum Start eines Programms an einer anderen Adresse; beide führen nach Beendigung dieses Programms wieder zu einer Rückkehr in das alte Programm. Die Rückkehradresse wird bei den Mikroprozessoren auf die gleiche Weise gespeichert (*Abstellen im Stack*), auch das Verschachteln (*Nesting*) wird durch die Verwendung des Stacks beim Interrupt genauso wie beim Unterprogrammaufruf durchgeführt.

Es bestehen zwischen Interrupts und Unterprogrammaufrufen natürlich auch *Unterschiede,* besonders weil

1. Unterprogramme vom Programmierer an ganz bestimmten Stellen des Programms aufgerufen werden, während der Interrupt durch äußere Signale jederzeit auftreten kann.
2. Die Sicherung von Inhalten der CPU nicht bei allen Mikroprozessoren in der gleichen Art beim Unterprogrammaufruf wie beim Interrupt erfolgt. Bei Prozessoren, z.B. 8080 bzw. 8085, bei denen bei einem Interrupt die gleichen Informationen sichergestellt werden wie bei einem Unterprogrammaufruf, erfolgt die Rückkehr aus einem Unterprogramm mit dem gleichen Befehl (z. B. RETURN) wie die Rückkehr nach einem Interruptprogramm. Bei einem Mikroprozessor wie dem MC 6800, bei dem bei einem Unterprogrammaufruf nur die Rückkehradresse, bei einem Interrupt aber noch weitere 5 Bytes sichergestellt werden, sind für die Rückkehr zwei verschiedene Befehle erforderlich (*Return from Subroutine:* RTS; *Return from Interrupt:* RTI).

## 5 Was ist Maskierung?

Nicht in jeder Situation ist es sinnvoll, daß ein Interruptsignal das laufende Programm unterbricht; in einer bestimmten Situation kann es auch sinnvoll sein, daß ein bestimmtes Interruptsignal wirksam wird, ein anderes aber nicht. Da der Prozessor keinen Einfluß auf die Bildung der Interruptsignale hat, muß er sich vor ihnen schützen können; dies geschieht durch **Maskenbits.** Ein Maskenbit besteht aus einem Flipflop: ist es gesetzt, so ist das Interruptsignal wirksam, ist es gelöscht, so kann das Interruptsignal nicht wirksam werden.

Die Maskenbits können sich direkt im Mikroprozessor-Chip befinden, wenn es mehrere Interrupteingänge gibt. So können mehrere Maskenbits erforderlich sein (z.B. 8085), die zu einer *Interruptmaske* zusammengefaßt werden. Die Interruptmasken, die sich direkt im Prozessor-Chip befinden, werden mit besonderen Befehlen gesetzt oder gelöscht, z. B. mit den Befehlen DI: *Disable Interrupt* (Sperre Interrupt) bzw. EI: *Enable Interrupt* (Lasse Interrupt zu) beim 8080.

**DMA:** *Direct Memory Access,* direkter Speicherzugriff. Schnellste Art, Daten in den Arbeitsspeicher zu laden, nämlich unter Umgehung der Steuereinheit.

**Analog-Digital-Wandler:** (ADC: *Analog-Digital Converter*). Rückwandlung digitaler Rechenwerte in analoge Ausgabedaten mit DAC: *Digital-Analog Converter.*

**Maskierung:** In der Daten- und Programmiertechnik meint man damit die Überlagerung zweier Binärworte derart, daß einzelne Bits „aus- oder eingeblendet" werden.

Beim 8085 gibt es neben diesen beiden Befehlen noch den Befehl SIM: *Setze Interrupt-Maske,* welcher gezielt bestimmte Interrupts maskieren kann.

Wird ein Interrupt ausgeführt, d.h. wird eine Interrupt Service Routine begonnen, so werden die im Prozessor befindlichen Interruptmasken meist automatisch gesperrt, damit nicht das soeben begonnene Programm sofort wieder unterbrochen wird. In der Interrupt Service Routine können natürlich Befehle vorhanden sein, die die Maske wieder setzen.

Die meisten Prozessoren verfügen auch über Eingänge für Interrupts, die nicht maskierbar sind, diese werden beim

MC 6800 und beim Z 80 als
NMI = **Non Maskable Interrupt**
beim 8085 als TRAP

bezeichnet. Der 8080 verfügt nicht über einen solchen Eingang. Diese Eingänge sind für Interruptursachen gedacht, die sinnvollerweise *immer* zur Erzeugung einer Programmunterbrechung führen sollen, z.B. beim Netzausfall, bei dem Informationen sichergestellt werden, ehe die Versorgungsspannung ganz zusammenbricht.

Auch die RESET-Eingänge, die das laufende Programm unterbrechen und ohne Speicherung einer Rückkehradresse wieder in das Betriebssystem zurückführen, sind nicht maskierbar.

Masken für Interrupts können außerhalb des Prozessors aus Registern aufgebaut werden; sie werden dann über E/A-Befehle gesetzt oder gelöscht. Auf diese Weise ist es möglich, in bestimmten Programmsituationen nur bestimmte Interrupts zuzulassen (**Bild 1**).

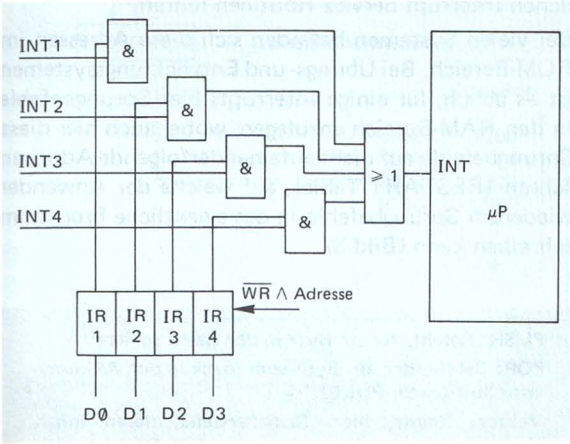



**Bild 1** Verwendung eines Maskenregisters außerhalb des Mikroprozessors

In den E/A-Bausteinen, die zu den Mikroprozessor-Familien gehören, z.B. PIA (MC 6800), PPS 8255 (8080), PIO (Z 80), RAM-E/A 8155 (8085), werden in bestimmten Betriebsarten Interruptanforderungen erzeugt, wobei durch E/A-Befehle programmiert wird, unter welchen Bedingungen diese Signale erzeugt werden bzw. ob sie zu unterdrücken sind. Auf diese Weise kann ein E/A-Vorgang über Steuer- und Quittungs-Signale (*Handshake-Modus*) abgewickelt werden.

## 6 Was muß bei einer Programmunterbrechung gerettet werden?

Wenn das Programm, welches unterbrochen wurde, später wieder fortgesetzt werden soll, so muß die Rückkehradresse, d.h. der aktuelle Stand des Befehlszählers, sichergestellt werden. Alle Interrupts mit Ausnahme des *Reset* werden erst wirksam, wenn ein Befehl abgearbeitet wurde, so daß der *Befehlszähler* (**PC**) bereits die Adresse des nächsten Befehls zeigt. Weiterhin ist es in den allermeisten Fällen notwendig, alle Registerinhalte der CPU sicherzustellen, da sonst eine sinnvolle Wiederaufnahme des alten Programms nicht möglich ist.

Die Rettung von Informationen geschieht teilweise automatisch, d.h. das Interruptsignal löst die dazu erforderlichen Vorgänge aus, teilweise muß die Rettung programmiert werden:

a) Beim MC 6800 werden sowohl der Befehlszähler (PC), der aus zwei Byte besteht, wie auch alle anderen Register im Prozessor (insgesamt 5 Byte) automatisch in den Hauptspeicher übertragen, wobei der Speicher vom Stack adressiert wird, der **Stackpointer** wird bei der Übertragung **dekrementiert.**

b) Beim 8080/8085 werden nur die beiden Bytes des Befehlszählers automatisch in den Stack übertragen, die übrigen Register können aber vom Programmierer durch einfache Stack-Befehle sichergestellt werden; dies muß am Beginn der Interrupt Service Routine geschehen (**Bild 2**).

**Non Maskable Interrupt:** (NMI). Interrupt-Anforderungen, die unbedingt und sofort nach Abschluß der gerade laufenden Instruktion akzeptiert werden muß.

**PC:** *Program Counter,* Programmzähler.

**Stackpointer:** Spezielles Register, das angibt, welche Speicherstelle im Stapelbereich (*Stack*) als nächste belegt werden kann.

**Dekrementierung:** Herunterzählung um eine Einheit (Hinaufzählen: *Inkrementierung*).

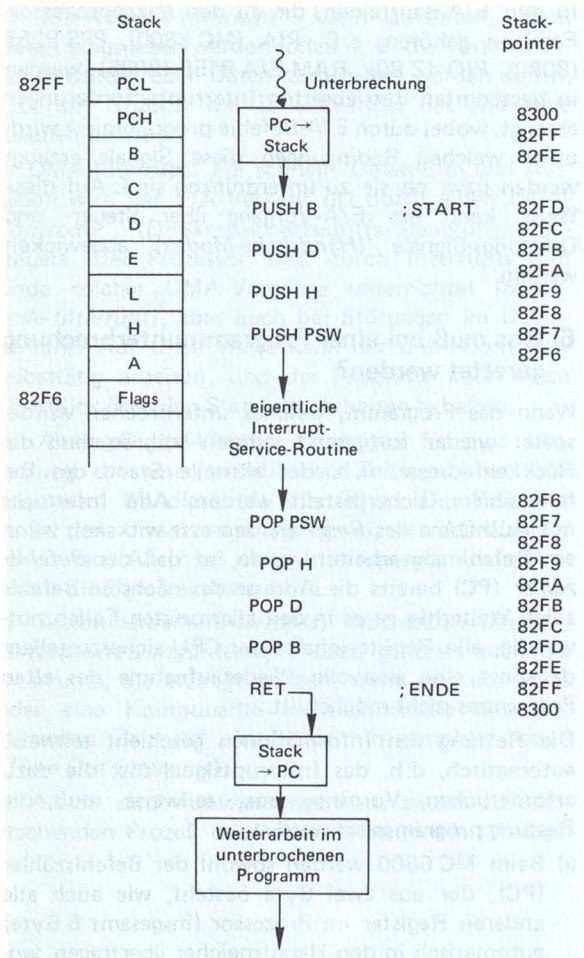



**Bild 2** Sicherung von CPU-Inhalten beim Interrupt (die Operationen PC → Stack und Stack → PC werden nicht programmiert, sondern bei Auftreten des Interrupts oder des RETURN-Befehls automatisch durchgeführt)

PCL: Program Counter Low;
PCH: Program Counter High;
PSW: Prozessor-Statuswort (Flags);
RET: RETURN

c) Der Z 80 kann ebenso wie der 8080 nach automatischer Rettung des Befehlszählers durch Stack-Befehle die Register in den Stack übertragen; er hat daneben aber auch die Möglichkeit, durch zwei Befehle innerhalb der CPU die Register sicherzustellen, da in dem Prozessorchip alle Register zweifach vorhanden sind. Es ist dabei allerdings zu beachten, daß diese Art der Rettung nur einmal erfolgen kann, da zu jedem Register nur eine Alternative vorhanden ist. Eine Verschachtelung von Interrupts (Abschnitt 9) ist nicht möglich. Es wird daher empfohlen, diese Art von Sicherstellung nur beim nichtmaskierbaren Interrupt durchzuführen.

## 7 Was ist ein Vektor bei der Interrupt-Bearbeitung

Die Reaktionszeit eines Systems auf den Interrupt hängt nicht nur davon ab, wie schnell ein Interruptprogramm gestartet wird, sondern auch davon, wie schnell das „richtige Programm", welches zur Bearbeitung dieses Interrupts dient, gestartet wird; der Interrupt selber muß bereits auf sein Programm „zeigen" oder einen **Vektor** bilden. Dazu gibt es bei den verschiedenen Mikroprozessoren verschiedene Möglichkeiten:

a) Das Gerät, welches den Interrupt erzeugt, bietet auf dem Datenbus eine Information an, welche zur Startadresse des Interruptprogramms führt. Beim 8080, dem 8085 und dem Z 80 (im Interruptmodus 0) ist dies eine Drei-Bit-Information, d. h. sie kann die Zahlen 0 bis 7 enthalten. Die Speicheradresse, bei welcher das Interruptprogramm beginnt, errechnet sich nach A = 8 X Information.

Wenn also die Information 101 = 5 angeboten wird, so startet das Programm auf der Adresse $40_D = 25_H$ (D: Dezimal, H: Hexadezimal).

Die einzelnen Startadressen sind damit nur 8 Bytes voneinander entfernt, damit ist es klar, daß hier nicht die eigentlichen Interrupt Service Routinen beginnen, sondern nur Sprungbefehle stehen, die auf die eigentlichen Interrupt Service Routinen führen.

Bei vielen Systemen befinden sich diese Adressen im ROM-Bereich. Bei Übungs- und Entwicklungssystemen ist es üblich, für einige Interrupts hier Sprungbefehle in den RAM-Bereich anzulegen, wobei auch hier diese Sprungbefehle auf dicht aufeinanderfolgende Adressen führen (RESTART-Table), auf welche der Anwender wiederum Sprungbefehle in das eigentliche Programm schreiben kann (**Bild 3**).

**PUSH:** Befehl, der ein Byte in den Stack schiebt.
**POP:** Befehl, der ein Byte vom Stack in den Akkumulator holt (auch: PULL).
**Vektor:** Zeiger, hier: Speicherstelle, deren Inhalt eine Adresse ist, also auf eine andere Speicherstelle „zeigt" (z. B. *Stackpointer*).

| Adresse | Maschinen-code | Label | Assemblercode | | |
|---|---|---|---|---|---|
| | | | ORG | 0028H | ROM-Bereich |
| 0028 | C3 C3 | RST5 | JMP | INT5 | |
| | 40 | | | | |
| | 83 | | | | |
| | 00 | | ORG | 0030H | |
| 0030 | C3 | RST6 | JMP | INT6 | |
| | 43 | | | | |
| | 83 | | | | |
| | 00 | | ORG | 0038H | |
| 0038 | C3 | RST7 | JMP | INT7 | |
| | 46 | | | | |
| | 83 | | | | |
| 8340 | C3 | INT5 | JMP | SERV5 | RAM-Bereich |
| | 00 | | | | |
| | 82 | | | | |
| 8343 | C3 | INT6 | JMP | SERV6 | |
| | 00 | | | | |
| | 83 | | | | |
| 8346 | C3 | INT7 | JMP | SERV7 | |
| | 60 | | | | |
| | 82 | | | | |

**Bild 3** Vektorisierung des Interrupts bei ROM-Belegung der Restart-Adressen. Das eigentliche Interruptprogramm für RST 5 beginnt auf 8200, für RST 6 auf 8300 und für RST 7 auf 8260, diese Festlegung kann vom Programmierer aber jederzeit durch Neubeschreiben des Speicherbereichs 8340 bis 8348 geändert werden.
(JMP: Jump, Springe).

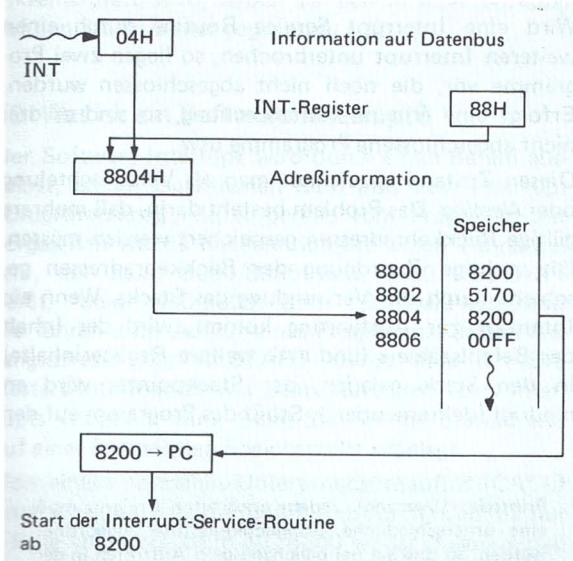



**Bild 4** Gewinnung der Startadresse für die Interrupt-Service-Routine bei Z 80 im Interrupt-Modus 2 (IM 2).

Beim Z 80 wird dieses System im Interruptmodus 2 verfeinert angewandt (der Interruptmodus beim Z 80 wird über eigene Befehle IMO = $ED46_H$, IM 1 = $ED56_H$, IM 2 = $ED5E_H$ festgelegt.). Das periphere Gerät bietet eine 8-bit-Information an, von der allerdings nur 7 bit ausgewertet werden. Im Mikroprozessor existiert ein Interrupt-Register, welches eine 8-bit-Information enthält. Diese stellen das höherwertige Byte eine Adresse dar, die vom peripheren Gerät gelieferten Bits stellen das niederwertige Byte dar. Die so gebildete Adresse ist nicht die Startadresse der Interrupt Service Routine, sondern sie enthält diese Startadresse (**Bild 4**). In diesem System können 128 Interrupt Service Routinen direkt angewählt werden.

b) Das Interruptsignal zeigt direkt auf die Startadresse des Interruptprogramms. Dies ist z.B. beim Z 80 im Interruptmodus 1 der Fall, in dem der IRQ eine Interrupt Service Routine auf der Adresse $38_H = 56_D$ startet, so funktioniert auch der nichtmaskierbare Interrupt NMI des Z 80, der ein Interruptprogramm auf der Adresse $66_H = 102_D$ startet.

Im 8085 gibt es vier Interrupteingänge, die direkt, ohne daß auf dem Datenbuch eine Information anstehen muß, zu einer bestimmten Startadresse führen, die Zuordnung ist

| | |
|---|---|
| TRAP (nicht-maskierbarer Interrupt) | Adresse: $24_H = 36_D$ |
| RST 5.5 | Adresse: $2C_H = 44_D$ |
| RST 6.5 | Adresse: $34_H = 52_D$ |
| RST 7.5 | Adresse: $3C_H = 60_D$ |

Die Startadressen sind also zwischen die Startadressen, die vom IRQ (Interrupt Request)-Signal und der Restart-Information auf dem Datenbus angewählt werden, geschoben, so daß z.B. die von der Restart-Information $110 = 6_D$ angewählte Adresse $30_H = 48_D$ mit 4 Bytes Abstand gefolgt wird von der Startadresse $34_H = 52_D$ des durch Signal RST 6.5 ausgelösten Interruptprogramms. Vier Bytes genügen aber zum Abspeichern eines Sprungbefehls, der nur 3 Bytes umfaßt.

c) Das Interruptsignal zeigt indirekt auf die Startadresse. Dies ist der Fall bei MC 6800, bei dem die 8 höchsten Speicheradressen belegt sind mit den vier Adressen der vier verschiedenen Interrupt Service Routinen, die von den vier Interruptursachen (IRQ-Signal, NMI-Signal, RESET-Signal, Software-Interrupt) gestartet werden.

**RST:** *Restart* (auch: RS). Definierter Neubeginn.

Die Vektorisierung des Interrupts dient also dazu, möglichst schnell ein Programm zu starten, welches für diese besondere Interruptursache zuständig ist. Der Interrupt soll auf die ihm zugehörige Routine zeigen.

## 8 Was ist eine Priorität und wie kann sie berücksichtigt werden?

In vielen Systemen gibt es mehrere Geräte oder Komponenten, die sich mit Interruptanforderungen bei der CPU melden können. Dabei kann es zu Konflikten kommen, wobei zwei Fälle zu unterscheiden sind:

a) Mehrere Interruptanforderungen liegen gleichzeitig an, sie werden zwar nicht gleichzeitig auftreten, aber nach Aufhebung der Maskierung oder nach Abarbeitung einer Instruktion verlangen sie gleichzeitig die Unterbrechung des Programms.

b) Ein laufendes Programm wird unterbrochen, die Interrupt Service Routine gestartet, kurz darauf kommt eine weitere Interruptanforderung.

Grundsätzlich ist eine Verschachtelung von Interruptprogrammen möglich (Abschnitt 9); aber eine fortlaufende Unterbrechung durch immer neue Interruptanforderungen ist nicht sinnvoll, da es so nie zur Beendigung einer Arbeit kommen kann. Die Unterbrechung eines Interruptprogramms kann aber nicht ausgeschlossen werden. So müssen ein Netzausfallinterrupt oder auch ein Uhr-Interrupt immer wirksam werden. Den Interruptursachen müssen daher verschiedene Prioritäten zugeordnet werden. Die Berücksichtigung dieser Prioritäten ist Aufgabe von CPU und anderen Komponenten des Systems.

Liegen mehrere Interruptanforderungen vor, die gleichzeitig bearbeitet werden wollen, so werden sie über ein **Prioritäts-Netzwerk** geführt. Dies kann in der CPU vorhanden sein (z. B. beim 8085 für TRAP, RST 5.5, RST 6.5, RST 7.5), man kann es selbst entwerfen oder fertige Chips verwenden. Die Arbeit eines dafür vorgesehenen Chips sei an einem Beispiel geschildert (INTEL 8214).

Dem Baustein werden 8 Interrupteingänge zugeführt, er verfügt über drei Ausgänge, welche die Restart-Information erzeugen; die Eingänge können 256 verschiedene Kombinationen einnehmen, die Ausgänge aber nur 8 verschiedene Kombinationen. Die Ausgangsinformation zeigt dabei immer die Information für den Interrupt der höchsten Priorität. Zeigt sie z. B. das Signal 110, so bedeutet dies, daß ein Interrupt der Priorität 6 vorliegt, kein Interrupt der Priorität 7. Ob Interrupts der Prioritäten 0–5 vorliegen, wird am Ausgang in diesem Fall nicht angezeigt.

Eine weitere Möglichkeit zur Bildung der Prioritäten besteht darin, daß der Prozessor eine Drei-Bit-Information an den 8214 gibt, welche anzeigt, auf welcher Interruptebene er gegenwärtig läuft. Wird ein Interruptprogramm durch einen Interruptwunsch der Priorität 5 angestoßen, so kann dieses Programm mit einem Ausgabebefehl gestartet werden, der die Information 5 zum 8214 gibt. Alle Interruptanforderungen dieser oder niedrigerer Priorität werden dann nicht mehr vom 8214 weitergeleitet.

Eine Priorität kann auch über eine Maskierung erreicht werden. Bei Start eines Interruptprogramms werden hardware-mäßig die Maskenbits gelöscht; innerhalb der Interrupt Service Routine können sie wieder gesetzt werden, so daß die Prioritäten dynamisch verwaltet werden können. Dies gilt auch für außerhalb des Prozessors liegende Masken.

Der Mikroprozessor verfügt nicht über Möglichkeiten, festzustellen, ob er in einem Interruptprogramm läuft und welche Priorität dieses hat. Nach Erfüllung des Interruptwunsches fährt er mit der Abarbeitung der Befehle fort, ohne noch feststellen zu können, daß dieses Programm durch einen Interrupt gestartet wurde.

## 9 Was ist die Verschachtelung von Interrupts?

Wird eine Interrupt Service Routine durch einen weiteren Interrupt unterbrochen, so liegen zwei Programme vor, die noch nicht abgeschlossen wurden. Erfolgt eine erneute Unterbrechung, so sind es drei nicht abgeschlossene Programme usw.

Diesen Zustand bezeichnet man als Verschachtelung oder *Nesting.* Das Problem besteht darin, daß mehrere gültige Rückkehradressen gespeichert werden müssen. Die richtige Zuordnung der Rückkehradressen geschieht durch die Verwendung des Stacks. Wenn ein Interrupt zur Ausführung kommt, wird der Inhalt des Befehlszählers (und evtl. weitere Registerinhalte) in den Stack geladen, der Stackpointer wird erniedrigt (dekrementiert). Stößt das Programm auf den

**Priorität:** (Vorrang). Jedem erwarteten Ereignis muß eine unterschiedliche Dringlichkeitsstufe zugeordnet werden, so daß sie bei gleichzeitigem Auftreten in der Reihenfolge ihrer Priorität abgearbeitet werden können.

Rückkehrbefehl aus dem Interruptprogramm, so wird die Information aus dem Stack in den Befehlszähler übernommen (und evtl. in andere Register) und der Stackpointer erhöht (inkrementiert). Auf diese Weise können beliebig viele Interruptprogramme ineinander verschachtelt werden, nur die Größe des Speichers begrenzt deren Anzahl.

Da Unterprogramme nach dem gleichen Schema ihre Rückkehradressen finden, können Interruptprogramme und Unterprogramme beliebig miteinander und ineinander verschachtelt werden. Dabei ist darauf zu achten, daß der Programmierer nicht innerhalb eines Interruptprogramms oder eines Unterprogramms den Inhalt des Stackpointers in unzulässiger Weise ändert. Das bedeutet nicht, daß innerhalb eines Interruptprogramms oder eines Unterprogramms der Inhalt des Stackpointers nicht geändert werden darf. Es muß aber dafür gesorgt sein, daß beim Auftreten des Befehls für Rückkehr aus dem Interruptprogramm (oder dem Unterprogramm) der Stackpointer wieder so steht wie beim Einstieg in das Programm. Wird der Stackpointer innerhalb des Programms umgeladen, so ist dieser Ladevorgang wieder rückgängig zu machen und der alte Zustand wiederherzustellen; wird der Stack als Datenspeicher genutzt, so sind in einem Programm genausoviel Speicher-Schreib-Befehle (z.B. PUSH oder PSH) wie Speicher-Lese-Befehle (z.B. POP oder PUL) aus dem Stack-Bereich zu programmieren. Da bei diesen Befehlen der Stackpointer dekrementiert bzw. inkrementiert wird, heben sie sich in ihrer Wirkung auf den Stackpointer gegenseitig auf.

## 10 Was ist ein Software-Interrupt?

Der Software-Interrupt wird durch einen Befehl ausgelöst, ist im eigentlichen Sinn also kein Interrupt (Unterbrechung), da er bereits vom Programmierer vorgesehen ist. Es handelt sich um einen 1-Byte-Befehl, der beim 8080/8085 bzw. Z 80 RESTART heißt, beim MC 6800 SWI: *Software-Interrupt.* Sie führen zum Aufruf eines Programms, dessen Anfangsadresse bei RESTART durch eine Drei-Bit-Restart-Information wie beim Auftreten eines Interrupts festgelegt wird. Beim SWI ist die Startadresse auf einer festgelegten Speicherzelle abgelegt.

Von einem normalen Unterprogrammaufruf (CALL) unterscheiden sich die Befehle dadurch, daß sie nur 1 Byte lang sind; beim MC 6800 weiter dadurch, daß sie nicht nur zu einer Speicherung der Rückkehradresse im Stack, sondern zu einer Speicherung aller Registerinhalte führen.

Fachraum-Mobiliar, Lehrgeräte, Literatur für alle Schularten
Unser Bildungsbeitrag zur Naturwissenschaft und Technik

**Mikrocomputertechnik für Schule und Beruf**

**Input: 575 00**
**Output: Wissensvorsprung**

LEYBOLD-HERAEUS hat in seinem neuen Mikrocomputer 575 00 den Mikroprozessor unabhängig gemacht. Für die experimentelle Einarbeitung bedeutet das: Jetzt beginnt die Einführung in die Mikrocomputertechnik eine Wissensstufe früher als bisher. Denn mit dem Mikroprozessor im Mikrocomputer kann auch unabhängig gearbeitet werden – unabhängig vom Monitorprogramm, vom Hauptspeicher und dessen Adressierung. Die Instruktionen werden dabei in zeitlicher Reihenfolge von Hand auf den Datenbus des Mikroprozessors gegeben und von diesem unmittelbar verarbeitet. Damit kann man bereits wichtige Kenntnisse über Struktur und Arbeitsweise vom Mikroprozessor mit dem Mikroprozessor allein erhalten.
Mit dem Aufbau eines Mikrocomputers geht es programmgemäß weiter.

Das Wichtigste in Daten:

| | |
|---|---|
| Mikroprozessor: | SAB 8080 A |
| ROM-Bereich: | 12 Bytes von Hand programmierbar<br>1024 Bytes SAB 2708 für Monitorprogramm<br>1024 Bytes auf Platine nachrüstbar |
| RAM-Bereich: | 256 Bytes SAB 2101 (2 x)<br>512 Bytes auf Platine nachrüstbar |
| Ein-/Ausgabekanäle: | je 1 x 8 Bit und 2 x 1 Bit, 4-mm-Buchsen, kurzschlußfest, überspannungssicher, Schmitttriggerverhalten, Pegelumschalter für 5 V/12 V, LED-Anzeige der logischen Belegung |
| Hexadezimale Anzeige: | 6-stellig, 20 mm Zeichenhöhe |
| Tastatur: | zur Eingabe von hexadezimalen Daten |
| Bandanschluß: | auch für automatisch ausgesteuerte Geräte |
| Stromversorgung: | 220 V (auf 110 V umsteckbar), 50/60 Hz |

Ist der Mikroprozessor erforscht, wird er schrittweise zu einem kompletten Mikrocomputer ausgebaut. Nach Adressierung und Anwahl von Speicherplätzen kann ein kurzes Programm oder ein einfaches Hilfsprogramm zur Eingabe selbst erstellt werden. Dann wird das Programm in einem von Hand programmierbaren Festwertspeicher fixiert und erprobt. Erst jetzt werden fest abgespeicherte Monitorprogramme angewählt.

Der LH-Mikrocomputer mit seinem unabhängigen Mikroprozessor und seiner guten Ausstattung (besonders mit den zahlreichen in 4-mm-Buchsen herausgeführten, elektronisch geschützten Ein-/Ausgabekanälen) gibt die Möglichkeit, EDV-Wissen vom ersten bis zum letzten Schritt zu vermitteln und einfache technische Anlagen zu steuern.

Fordern Sie mehr Information.

WP-40-1278

Ein geübter Programmierer schreibt kürzere Programme sicher „aus dem Handgelenk" hin. Vor der Entwicklung von umfangreichen Programmen wird jedoch eine grafische „Zerlegung" des Problems unumgänglich sein. Dafür bieten sich heute im wesentlichen zwei Verfahren an. Die Unterschiede sowie Vor- und Nachteile werden im folgenden Beitrag herausgearbeitet und mit Beispielen illustriert.

W. Schneider

# Grafische Darstellung von Programmabläufen

Vor der Programmierung eines Problems in einer beliebigen Programmiersprache empfiehlt es sich,

- das Problem aufzubereiten und
- den Programmablauf grafisch darzustellen.

Erst anschließend sollte man, zumindest bei umfangreichen Problemen, zum Schreiben des **Primärprogrammes** in einer beliebigen Programmiersprache übergehen.

## 1 Problemaufbereitung

Zur Problemaufbereitung gehört

- eine vollständige Formulierung der Aufgabe und
- eine Problemanalyse der Aufgabe.

Die Aufgabe ist zunächst vollständig mit allen **Randbedingungen** in der Umgangssprache zu formulieren. In der darauf folgenden **Problemanalyse** ist u.a. zu untersuchen,

- ob die Aufgabe überhaupt mit Hilfe einer Datenverarbeitungsanlage (kurz DVA) gelöst werden kann,
- welche alternativen Lösungswege sich für die Aufgabe anbieten und
- welcher der möglichen Lösungswege der günstigste ist.

## 2 Darstellung von Programmabläufen

Nachdem bei der Problemaufbereitung ein günstig erscheinender Lösungsweg gefunden wurde, empfiehlt es sich vielfach, die einzelnen Schritte zur Lösung des Problems grafisch darzustellen. Hier stehen dem Programmierer zwei wichtige Darstellungsweisen zur Verfügung:

- Darstellung mit Hilfe von *Programmablaufplänen.*
- Darstellung mit Hilfe von *Struktogrammen.*

Diese beiden Darstellungsweisen sollen im folgenden besprochen und miteinander verglichen werden.

### 2.1 Programmablaufpläne

Ein Programmablaufplan ist eine grafische Darstellung, die den Arbeitsablauf einer Problemstellung in einzelnen Schritten darstellt.

Die Programmablaufpläne setzen sich aus verschiedenen Sinnbildern zusammen, die nach DIN 66001 genormt sind. Durch Einfügen eines Textes in die Sinnbilder wird die Art der Vorgänge genau spezifiziert. Die Reihenfolge der Vorgänge wird durch Pfeile angedeutet. Für einfache Aufgaben genügt die Kenntnis der in **Bild 1** aufgeführten Sinnbilder.

**Primärprogramm:** (*Quellenprogramm, Source Program*). In einer höheren Sprache oder in *Assembler* geschriebenes Programm, das noch in die Maschinensprache übersetzt werden muß.

**Randbedingung:** Begriff aus der Mathematik, der allgemein verwendet wird, um das Verhalten eines Systems an definierten Bereichsgrenzen zu beschreiben, z.B. dort, wo ein Parameter null wird oder wo das System aufhört zu existieren.

**Problemanalyse:** Systematische Untersuchung eines Problems zur Ermittlung charakteristischer Größen und Verhaltensweisen.

| Sinnbild | Bedeutung | Beispiele |
|---|---|---|
| Text | allgemeine Operation | C = A – B; schließe Ventil A |
| Text (ja / nein) | Verzweigung | A – B < 0 ? (ja / nein); Ventil A geschlossen? (ja / nein) |
| | **Bemerkung:** Der Text muß eine Frage (Bedingung) enthalten, die entweder mit ja oder nein zu beantworten ist. Je nach Beantwortung der Frage wird das Programm mit dem „Ja"- oder „Nein"-Zweig fortgesetzt. | |
| Text | Eingabe Ausgabe | Lies A; Drucke B |
| Text | Grenzstelle | STOP; START |
| Text | **Bemerkung:** Die Grenzstelle ist das Sinnbild für den Beginn oder das Ende eines Programmes | |
| n | Übergangsstelle | 8; 8 |
| n | **Bemerkung:** Falls ein Programmablaufplan auf einem anderen Blatt fortgesetzt werden muß, kennzeichnen die Übergangsstellen mit gleichen Zahlen, an welcher Stelle das Programm auf dem anderen Blatt fortgesetzt werden muß. | |
| | Richtungspfeil | **Bemerkung:** Falls keine Richtungspfeile angegeben werden, wird der Programmablaufplan von oben nach unten und von links nach rechts gelesen. |

**Bild 1** Nach DIN 66001 genormte Sinnbilder von Programmablaufplänen

Das Zeichnen der Programmablaufpläne wird durch Schablonen erleichtert **(Bild 2)**.

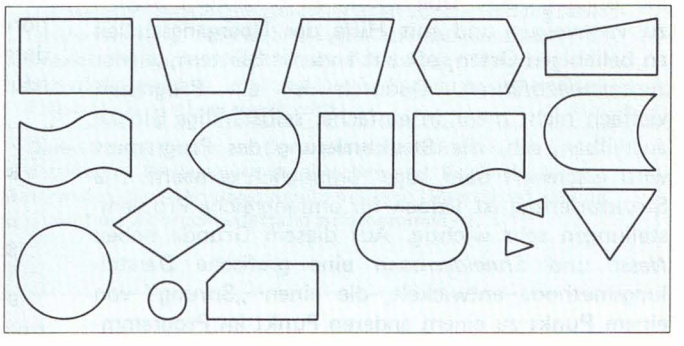


**Bild 2**
Beispiel einer Schablone zur Erstellung von Programmablaufplänen

**Vorteile bei der Anwendung von Programmablaufplänen**

Der Programmablaufplan erweist sich bei umfangreichen Aufgaben als sehr zweckmäßig. Insbesondere sind folgende Vorteile zu nennen:

- Der Programmablaufplan verschafft dem Programmierer durch die logische Gliederung einen Überblick über den Gang der Rechnung.
- Der Programmablaufplan verhindert das Programmieren von „Sackgassen".

  Eine Sackgasse würde sich z.B. in einem Programm ergeben, wenn ein Zweig einer Verzweigung durch Vergeßlichkeit des Programmierers nicht weiter berücksichtigt würde. Wenn bei einer späteren Benutzung des Programms dieser Zweig gewählt wird, so gibt es keine darauffolgende Anweisung. Das Programm endet in einer Sackgasse. In einem Programmablaufplan sind derartige Sackgassen gut zu erkennen und können somit vermieden werden.
- Der Programmablaufplan bietet ein gutes Verständigungsmittel zwischen einem Spezialisten und einem Programmierer.

  Spezialisten verfügen teilweise über keine ausreichenden Programmierkenntnisse. Ein Programmierer hingegen verfügt nicht immer über die notwendigen Spezialkenntnisse, um programmierbare Regeln aus einer Aufgabenstellung abzuleiten. Hier bietet sich der Programmablaufplan als gemeinsames Verständigungsmittel an.
- Der Programmablaufplan hilft bei der Fehlersuche von logischen Fehlern.

  Durch die logische Gliederung des Problems in eine Folge von Einzelschritten ist der Programmablaufplan wegen der besseren Übersicht meist besser als das Programm selbst geeignet, logische Fehler im Programmablauf zu finden.
- Der Programmablaufplan dient zur Dokumentation des Programms.

  Programme sollen auch später, eventuell von anderen Personen, wieder benutzt werden können. Sie müssen sich auf einfache Art darüber informieren können, wie das Programm aufgebaut ist, welcher Lösungsweg gewählt wurde usw. In vielen Fällen kann der Programmablaufplan eine spezielle Programmbeschreibung ersparen.

## 2.2 Struktogramme

Programmablaufpläne sind als Hilfsmittel zur grafischen Darstellung von Programmen weit verbreitet und auch genormt. Sie beschreiben die logische Struktur eines **Algorithmus** und den Ablauf, d.h. die Reihenfolge der Schritte zu programmierender Probleme. Die Praxis zeigt jedoch, daß dieses Hilfsmittel den Programmierer dazu verführt, Programme ohne größere Überlegung an beliebigen Stellen zu *verzweigen* und mit Hilfe der Übergangsstellen an beliebigen Orten, oft auf anderen Blättern, wieder *zusammenzuführen.* Dadurch ist ein Programm vielfach nicht mehr in einfache, selbständige Blöcke aufteilbar, d.h. die **Strukturierung** des Programms wird erschwert oder sogar unmöglich gemacht. Die Strukturierung ist jedoch für umfangreiche Problemstellungen sehr wichtig. Aus diesem Grunde haben *Nassi* und *Shneidermann* eine grafische Darstellungsmethode entwickelt, die einen „Sprung" von einem Punkt zu einem anderen Punkt im Programmablaufplan verhindert. Damit wird gleichzeitig sichergestellt, daß nicht nur der „*Programmfluß*", sondern auch die „*Programmstruktur*" deutlich wird, d.h. die Bedeutung der einzelnen Programmteile für den Gesamtablauf wird auf den ersten Blick erkennbar.

Das Lesen und Zeichnen von Struktogrammen ist schnell erlernbar, da im wesentlichen nur 4 Basis-Symbole verwendet werden, die in speziellen Fällen etwas verändert werden. Sie sind zwar nicht genormt, jedoch in der Literatur bislang gleichartig dargestellt worden. Die 4 Basis-Symbole und zwei häufig benutzte Varianten zeigen **Bild 3** und **Bild 4**.

**Algorithmus:** Rechenvorschrift, festgelegtes Schema für einen Lösungsweg. Hier durch Programmablaufpläne dargestellt.

**Strukturierung:** Zerlegung eines Problems oder eines Textes in leicht übersehbare, logisch zusammenhängende Bestandteile.

| Sinnbild | Bedeutung | Erläuterung |
|---|---|---|
| Text | *Prozeß* (Aktivität, Operation) | Das Prozeßsinnbild dient zur Darstellung eines oder mehrerer Befehle wie z.B.<br>• Wertzuweisungen (arithmetische Zuordnungsanweisungen)<br>• Ein- und Ausgabeanweisungen (diese besitzen in Programmablaufplänen ein spezielles Sinnbild)<br>• Unterprogrammaufrufe (diese besitzen in Programmablaufplänen ebenfalls ein spezielles Sinnbild, das allerdings in Bild 1 nicht aufgenommen wurde).<br>Die Form des Prozeßsinnbildes ist rechteckig. Die Größe ist frei wählbar. |
| Text<br>F T | *Verzweigung* (Entscheidung, Selektion) | Das Verzweigungssinnbild dient zur Darstellung bedingter Verzweigungen mit zwei Alternativen (**Ja/Nein-Entscheidung**). Das Verzweigungssinnbild besteht aus drei Dreiecken. Das mittlere Dreieck (Text) enthält die Bedingung (Frage), die entweder mit NEIN (F = FALSE) oder JA (T = TRUE) zu beantworten ist. Je nach Beantwortung der Frage wird das Programm mit einem Prozeßsinnbild, das direkt auf das linke Dreieck (F) oder auf das rechte Dreieck (T) folgt, fortgesetzt.<br>Die Größe des Verzweigungssinnbildes ist von der jeweiligen Anwendung und dessen Erfordernissen abhängig. |
| Wiederholungsbed.<br>Rumpf | *Wiederholung* (Schleife, Iteration) | Das Wiederholungssinnbild dient zur Darstellung von Schleifen.<br>Die Wiederholungsbedingung steht links oben im Wiederholungssinnbild. Hier wird z.B. die Zahl der Wiederholungen angegeben oder die Bedingung, unter der die Wiederholung zu beenden ist.<br>Die zu wiederholenden Anweisungen stehen im inneren Rechteck (Rumpf). Der Rumpf kann aus einer Struktur beliebiger Verwicklung bestehen.<br>Eine Verschachtelung von Schleifen ist möglich (weitere Schleifen im Rumpf!). |
| BEGIN<br>Rumpf<br>END | *Anfang und Ende* | Das Anfang- und Ende-Sinnbild dient zur Darstellung des Beginns oder des Endes von Programmen. |

**Bild 3** Grund-Symbole für Struktogramme

| Sinnbild | Bedeutung | Erläuterung |
|---|---|---|
| Text<br>1 2 3 4 | *Mehrfachverzweigung* (Schalter) | Das Mehrfachverzweigungssinnbild dient zur Darstellung bedingter Verzweigungen mit mehr als zwei Alternativen. Das obere Dreieck des Sinnbildes enthält die Fallabfrage (Bedingung), d.h. hier wird angegeben, unter welcher Bedingung zu den einzlnen Fällen (1, 2, ... n) verzweigt wird. |
| Wiederholungsbed.<br>Rumpf<br>Abbruchbed. | *Schleife mit Abbruchbedingung* | Dieses Sinnbild dient zur Darstellung von Schleifen, die unter bestimmten Bedingungen abzubrechen sind. Es gleicht dem normalen Wiederholungssinnbild mit dem Unterschied, daß im Rumpf die Abbruchbedingung aufgenommen ist. |

**Bild 4** Häufig benutzte Symbolvarianten für Struktogramme

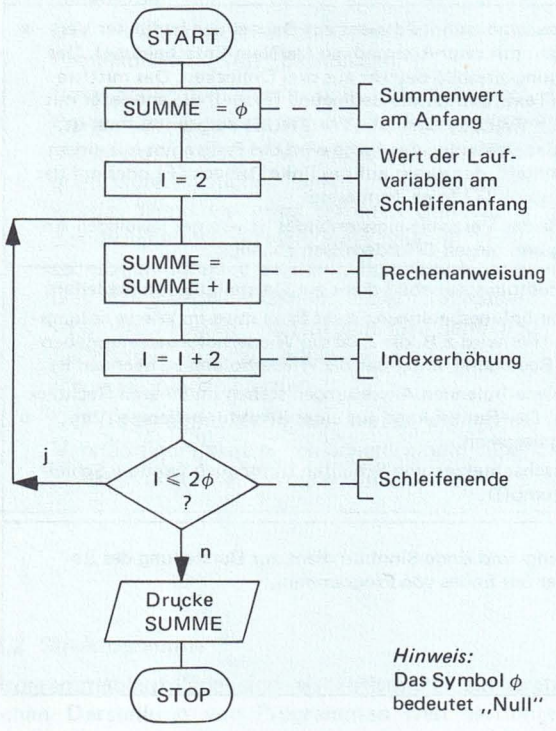
## 2.3 Beispiele

### Beispiel 1

*Aufgabenstellung:*

Es soll die Summe der ganzen geraden Zahlen von 1 bis 20 gebildet werden. Geben Sie zur Lösung dieses Problems sowohl einen Programmablaufplan als auch ein Struktogramm an.

*Programmablaufplan:*



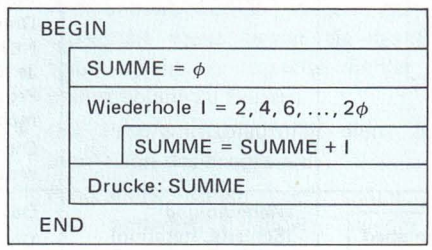

*Erklärung:*

Zunächst müssen die Anfangswerte gesetzt werden. So wird zunächst der Inhalt der Speicherzelle, in der die *Summen* nach jedem Schleifendurchlauf abgespeichert werden, *Null* gesetzt, damit Werte, die vorher möglicherweise in der Speicherzelle standen, das Ergebnis nicht verfälschen können (Summe = ϕ).

Außerdem wird der Anfangswert der Laufvariablen I auf den ersten ganzzahligen Wert, der zur Summe beiträgt, festgelegt (I = 2).

Es erfolgt anschließend die erste Rechnung: SUMME = ϕ + 2. Darauf folgt die erste Indexerhöhung, die sich aus I = I + 2 mit den vorgegebenen Werten zu I = 2 + 2 = 4 ergibt. Durch eine Abfrage, ob I ≤ 2ϕ ist, wird der **Index** I überprüft. Ist der Wahrheitswert wahr, wird die zweite Rechnung SUMME = 2 + 4 ausgeführt usw.. Die Schleife wird so lange durchlaufen, bis I > 20 wird. Dann wurden alle ganzen geradzahligen Zahlen einschließlich 20 aufaddiert und die Summe kann gedruckt werden.

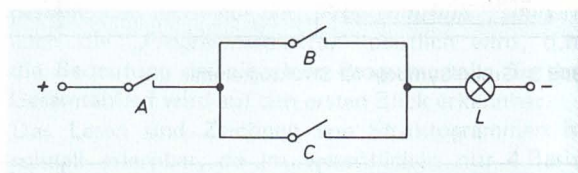
*Struktogramm:*




### Beispiel 2

*Aufgabenstellung:*

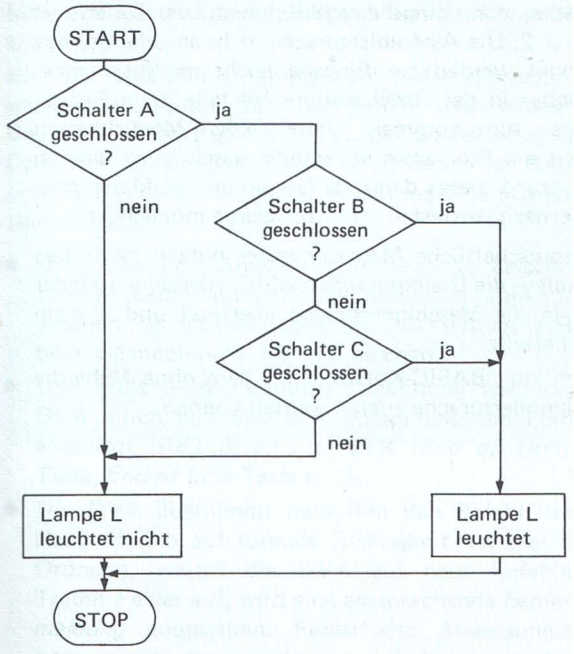
Es soll die Ablaufstruktur folgender Lampenschaltung grafisch mit Hilfe eines Programmablaufplanes und eines Struktogrammes dargestellt werden, die die Lampe zum Leuchten bzw. nicht zum Leuchten bringt.



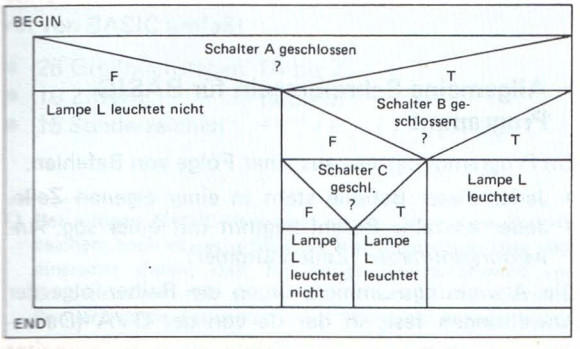

**Ja-Nein-Entscheidung:** Zwingende Auswahl zwischen zwei verschiedenen Zuständen (*Binärzustände*), auch logische Entscheidung genannt.

**Index:** Kennzeichen, Meßzahl, Laufzahl, Unterscheidungsangabe (Plural: *Indizes*). Die Ausführung einer Kennzeichnung wird *Indizierung* genannt.

*Programmablaufplan:*




*Struktogramm:*

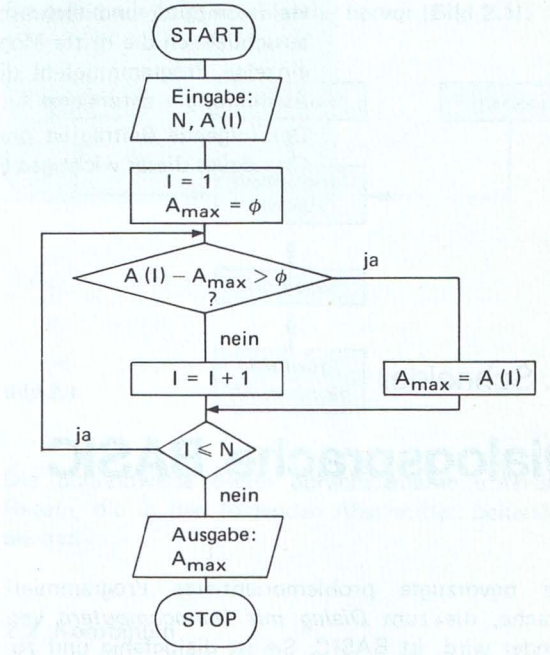



**Beispiel 3**

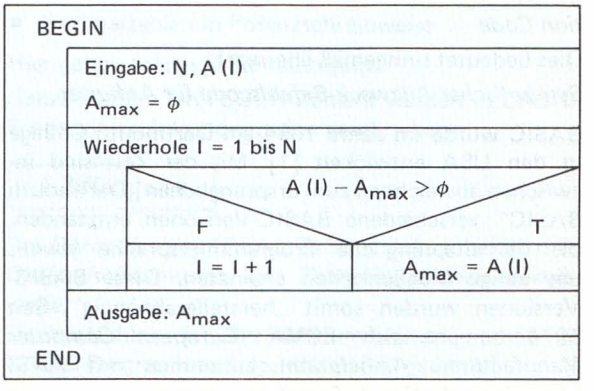
*Aufgabenstellung*

Aus einer bestimmten Anzahl N von positiven Zahlen A (I) soll durch einen Suchvorgang die maximale Zahl Amax ermittelt werden. Die Ablaufstruktur soll mit Hilfe eines Programmablaufplanes und eines Struktogrammes dargestellt werden.

*Programmablaufplan:*




*Struktogramm:*



Für den Umgang mit Mikrocomputern sind drei Programmiersprachen von Bedeutung: 1. Die Maschinensprache, d.h. Eingabe von Befehlen und Daten mit hexadezimal codierten Bitmustern; 2. Die Assemblersprache, d.h. anstelle der nur schwer lernbaren Hexadezimalcodes werden zur Eingabe leicht merkbare mnemonische Ausdrücke benutzt, wobei in der Regel weitere Vorteile beim Fehlersuchen, Ausbessern und Ergänzen hinzukommen. Diese beiden Möglichkeiten haben gemeinsam, daß immer erst ein Programm vollständig geschrieben werden muß, im Falle der Assembler-Nutzung dieses dann als Ganzes in die Maschinensprache übersetzt wird und erst hiernach Austesten und Korrigieren möglich ist.

Heimcomputer und technisch-wissenschaftliche Mikrocomputer nutzen heute fast ausschließlich die dritte Möglichkeit – die Dialogsprache BASIC. Hierbei wird jeder einzelne Programmbefehl direkt in die Maschinensprache übersetzt und – nach Ausführung – sofort eine Antwort erteilt.

Der folgende Beitrag ist praktisch ein „BASIC-Kursus", mit dem ohne Mühe die Grundzüge dieser wichtigen Programmiersprache erlernt werden können.

W. Schneider

# Dialogsprache BASIC

Die bevorzugte problemorientierte Programmiersprache, die zum *Dialog mit Heimcomputern* verwendet wird, ist BASIC. Sie ist dialogfähig und zudem noch relativ schnell und leicht erlernbar. Darauf weist schon der Name der Programmiersprache hin:

BASIC, d.h. *Beginners All-purpose Symbolic Instruction Code*

Dies bedeutet sinngemäß übersetzt:

*Symbolischer Allzweck-Befehlscode für Anfänger*

BASIC wurde im Jahre 1964 am Dartmouth College in den USA entwickelt [1]. Mit der Zeit sind inzwischen abweichend zum ursprünglichen „Dartmouth BASIC" verschiedene BASIC-Versionen entstanden, die die ursprüngliche Programmiersprache jeweils um einige Möglichkeiten ergänzten. Diese BASIC-Versionen wurden somit „herstellerabhängig". Seit 1974 bemüht sich ECMA (*European Computer Manufacturers Association*) zusammen mit ANSI (*American National Standards Institute*) um eine Standardisierung von BASIC. Im Dezember 1976 wurde eine Norm für ein sog. „Minimal BASIC" vorgeschlagen [2, 3]. Die Hersteller von Heimcomputern halten sich mehr oder weniger an diesen Vorschlag. Der BASIC-Befehlsvorrat, auf den in diesem Beitrag näher eingegangen wird, wurde so ausgewählt, daß er sowohl in dem Normvorschlag als auch in allen modernen BASIC-Versionen der Heimcomputer-Hersteller vorhanden ist und sich damit auf ein Mindestmaß an Befehlen beschränkt.

## 1 Allgemeine Schreibregeln für BASIC-Programme

Ein Programm besteht aus einer Folge von Befehlen.

- Jeder dieser Befehle steht in einer eigenen Zeile.
- Jeder einzelne Befehl beginnt mit einer sog. *Anweisungsnummer* (Zeilennummer).

Die Anweisungsnummern legen die Reihenfolge der Anweisungen fest, in der sie von der DVA (Datenverarbeitungsanlage, hier: Heimcomputer) bearbeitet werden. Aus diesem Grunde muß das Programm nicht unbedingt in seiner logischen Reihenfolge eingegeben werden. Es empfiehlt sich, bei der Ersteingabe des

Programms eine größere Schrittweite bei den Anweisungsnummern zu wählen, um später noch Befehle einfügen zu können, falls sich dies als notwendig erweist.

Auf die Anweisungsnummer folgt die *Anweisung* selbst. Sie beginnt stets mit einem *Schlüsselwort.* Das Schlüsselwort gibt den Typ der auszuführenden Operation an. Die Schlüsselworte werden in der Regel noch durch nähere Angaben zu den speziellen Operationen ergänzt.

- Die Länge einer Anweisung ist durch die Zeichenzahl je Zeile begrenzt. Die maximale Zahl der Zeichen ist herstellerabhängig. Sie beträgt z. B. beim Heimcomputer PET 40 Zeichen.
- Das Ende jeder einzelnen Anweisung wird der DVA durch Betätigen einer entsprechenden Taste angezeigt (RETURN-Taste, ETX (*End of Text*)-Taste, *End of Line*-Taste o. ä.).
- Die DVA übernimmt daraufhin den Befehl und überprüft ihn auf formale Richtigkeit. Ist alles in Ordnung, wartet die DVA auf neue Befehle. Treten Fehler auf, wird eine entsprechende *Fehlermeldung* ausgegeben. Fehlerhafte Anweisungen kann man ersetzen, indem man die korrigierte Anweisung mit der gleichen Anweisungsnummer wie die fehlerhafte Anweisung erneut eingibt. Vielfach gibt es jedoch auch Möglichkeiten, die fehlerhafte Stelle direkt zu korrigieren.

## 2 BASIC-Sprachelemente

### 2.1 BASIC-Zeichenvorrat

BASIC ist eine Programmier*sprache.* Wie jede andere Sprache besitzt auch sie einen begrenzten Zeichenvorrat. Der in diesem Beitrag verwendete Zeichenvorrat von BASIC umfaßt

- 26 Großbuchstaben (A bis Z)
- 10 Ziffern (Ø bis 9)
- 16 Sonderzeichen + − * / = , . ( ) ' : ; < > " ↑ [1]

[1]) Bei einigen BASIC-Versionen ist der Vorrat an Sonderzeichen noch etwas größer als hier angegeben. Dies liegt einerseits daran, daß für einige Kombinationen von Sonderzeichen mit spezieller Bedeutung ein spezielles Sonderzeichen verwendet wird (Beispiel: # anstelle von < >). Andererseits liegt es daran, daß BASIC in diesem Beitrag nur einführend behandelt wird und spezielle Themen ausgeklammert wurden. So benötigt man z. B. zur Kennzeichnung von sog. „Stringvariablen" das Sonderzeichen $.

Wie in einer normalen Sprache, in der aus dem Zeichenvorrat Worte und Sätze gebildet werden, können auch in der Programmiersprache BASIC entsprechende Sprachelemente gebildet werden. Da BASIC mathematisch-naturwissenschaftlich orientiert ist, gehören zu den Sprachelementen insbesondere

- Konstanten
- Variablen
- Funktionen
- Operationszeichen
- Ausdrücke
- Anweisungen

Sie gehen wie folgt auseinander hervor (**Bild 2.1**):

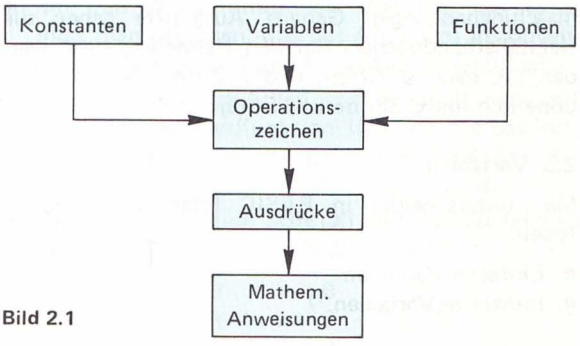



**Bild 2.1**

Die Schreibweise dieser Sprachelemente unterliegt Regeln, die in den folgenden Abschnitten behandelt werden.

### 2.2 Konstanten

Man unterscheidet folgende Arten von Konstanten

- Ganze Zahlen
- Dezimalzahlen in Dezimalschreibweise
- Dezimalzahlen in Potenzschreibweise

Hier gelten folgende Schreibregeln:

*Ganze Zahlen* (INTEGER-Zahlen) werden in BASIC-Programmen folgendermaßen geschrieben:

| ± Ziffernfolge |
|---|

Bei *Dezimalzahlen* (REAL-Zahlen) unterscheidet man *Dezimalschreibweise* und *Zehnerexponentialschreibweise.*

Dezimalzahlen in *Dezimalschreibweise* werden in BASIC-Programmen folgendermaßen geschrieben:

| ± Ziffernfolge . Ziffernfolge |
|---|

Als Dezimalzeichen ist nur der Dezimalpunkt erlaubt.

Dezimalzahlen in *Zehnerexponentialschreibweise* werden in BASIC-Programmen folgendermaßen geschrieben:

| ± Ziffernfolge . Ziffernfolge E ± Exponent |
|---|

Im Gegensatz zur Mathematik muß vor dem Exponenten einer BASIC-Konstanten in Zehnerexponentialschreibweise mindestens eine Ziffer stehen. Der Exponent muß immer eine ganze Zahl sein. In der Schreibweise wirkt sich vereinfachend aus, daß führende und nachgestellte Nullen ohne Bedeutung sind. Die maximale Größe der Ziffernfolge ist maschinenabhängig. Genaue Auskünfte geben die Herstellerhandbücher. Bei den Heimcomputern werden i.a. max. 9 Ziffern und 2 Ziffern bei dem Exponenten (max. 38) berücksichtigt.

### 2.3 Variablen

Man unterscheidet in BASIC folgende Variablentypen:

- Einfache Variablen
- Indizierte Variablen [1])

*Einfache Variablen*

Eine Variable steht stellvertretend für einen momentanen Wert.

Den variablen Größen werden dazu symbolische Namen gegeben. Sie dienen der Unterscheidung der Variablen voneinander und dürfen daher nur eindeutig verwendet werden.

Ein BASIC-Variablenname wird gebildet:

- aus 1 bis 2 BASIC-Zeichen
- das erste Zeichen muß ein Buchstabe sein
- das zweite Zeichen muß, falls es benutzt wird, eine Ziffer sein.

*Beispiel:* A, B1

*Indizierte Variablen*

Die Indizierung bietet die Möglichkeit, Variablen zu beziffern. Bei der Indizierung in BASIC sind folgende Regeln zu beachten:

- Der Variablenname darf bei indizierten Variablen i.a. nur aus einem einzelnen Buchstaben bestehen.
- Der dem Variablennamen folgende Index muß in Klammern gesetzt werden.
- Die indizierte Variable darf höchstens zwei Indizes besitzen. Beide Indizes müssen durch ein Komma voneinander getrennt werden.
- Der Index kann eine Zahl, eine Variable, eine indizierte Variable oder ein arithmetischer Ausdruck sein.

*Beispiel:* A(5), B(I), Z(N + 1)

*Feld und Feldvereinbarungen*

Die Menge aller indizierten Variablen, die den gleichen Variablennamen haben, werden als Feld bezeichnet. Man unterscheidet in BASIC ein- und zweidimensionale Felder, je nach Anzahl der Indizes.

Ein zweidimensionales Feld (Tabelle, Matrix) besteht aus waagerechten Zeilen und senkrechten Spalten. Der erste der beiden Indizes gibt die Zeilennummer, der zweite die Spaltennummer an.

Besteht ein Feld nur aus einer Zeile oder Spalte, so handelt es sich um ein eindimensionales Feld (Liste, Vektor).

Jedes Element eines Feldes kann durch Angabe der Feldvariablen und durch Nennung der jeweiligen Indizes angesprochen werden.

Mit Hilfe der DIM-Vereinbarung wird der DVA mitgeteilt, wieviel Speicherplatz für jedes Feld bereitgestellt werden muß.

Für eindimensionale Felder ist die allgemeine Form der DIM-Vereinbarung:

| n DIM $V_1(m_1), V_2(m_2), \ldots, V_n(m_n)$ |
|---|

$V_n$ Feldname
$m_n$ Indexmaximalwert

Für zweidimensionale Felder ist die allgemeine Form der DIM-Vereinbarung

| n DIM $V_1(m_{1Z}, m_{1S}), V_2(m_{2Z}, m_{2S}), \ldots, V_n(m_{nZ}, m_{nS})$ |
|---|

$V_n$ Feldname
$m_{nZ}$ Indexmaximalwert der Zeile
$m_{nS}$ Indexmaximalwert der Spalte

Die Feldvereinbarung wird i.a. an den Programmanfang gestellt.

Vielfach kann die DIM-Vereinbarung auch entfallen, wenn eine bestimmte Anzahl von Feldelementen nicht überschritten wird. Dies ist abhängig vom jeweiligen Heincomputer und muß dem Handbuch entnommen werden.

[1]) Ein weiterer Variablentyp, die sog. Stringvariablen, werden in diesem Beitrag nicht behandelt.

*Beispiel:*

3 DIM A(5Ø), B(1Ø, 15)

Für die indizierte Variable A werden 5Ø Speicherplätze reserviert.

Für die indizierte Variable B werden 1Ø * 15 = 150 Speicherplätze reserviert.

### 2.4 Arithmetische Operationszeichen

Operationszeichen geben die auszuführende mathematische Operation an. Durch sie wird das Rechenwerk der DVA zu bestimmten Rechenschritten veranlaßt.

BASIC kennt folgende arithmetische Operatoren:

| Operation | BASIC | Beispiel |
|---|---|---|
| Addition | + | A + B |
| Subtraktion | – | A – B |
| Multiplikation | * | A * B |
| Division | / | A / B |
| Potenzbildung | ↑ | A ↑ B |

Andere arithmetische Operationen wie z. B. das Wurzelziehen und das Logarithmieren, werden nicht durch derartige Sonderzeichen ausgedrückt, sondern mit Hilfe sogenannter Standardfunktionen, auf die im folgenden Abschnitt eingegangen wird.

### 2.5 Standardfunktionen

Um technisch-mathematische Probleme lösen zu können, werden gewisse Standardfunktionen, wie z. B. sin, cos, log usw. benötigt. Eine Reihe dieser Standardfunktionen liegen in einer DVA fest programmiert vor und können durch Nennung ihres Namens aufgerufen und im Programm wie Variablen benutzt werden.

In BASIC stehen standardmäßig i. a. folgende Funktionen zur Verfügung:

| Übliche Schreibweise | Bedeutung | BASIC |
|---|---|---|
| $\sqrt{x}$ | Quadratwurzel | SQR (X) |
| $e^x$ | Exponentialfunktion | EXP (X) |
| ln x | Natürl. Logarithmus | LOG (X) |
| sin α | Sinus | SIN (A) |
| cos α | Cosinus | COS (A) |
| tan α | Tangens | TAN (A) |
| arc tan α | Arcustangens | ATN (A) |
| \|x\| | Absolutbetrag | ABS (X) |
| [x] | Ganzzahliger Anteil von x | INT (X) |
| sgn x | Signum von x | SGN (X) |
| | Zufallszahl zwischen 0 und 1 | RND (X) |

Auf den Namen der Standardfunktion folgt, durch Klammern getrennt, das Argument der Standardfunktion. Es darf aus beliebigen arithmetischen Ausdrücken, Variablen, Konstanten oder Standardfunktionen bestehen.

Einige Standardfunktionen sollen im folgenden noch erläutert werden.

- Die Quadratwurzel ist nur für positive Werte definiert, da sich sonst imaginäre Werte ergeben.
- Der natürliche Logarithmus ist ebenfalls nur für positive Werte definiert.
- Bei den Winkelfunktionen, wie sin, cos usw., muß man beachten, daß das Argument im Bogenmaß eingesetzt wird und *keinesfalls* im Gradmaß.
- Den ganzzahligen Anteil von X erhält man mit Hilfe der Standardfunktion INT (X). Sie bewirkt, daß die größte *ganze* Zahl (daher INTEGER), die kleiner oder gleich X ist, ermittelt wird.
- Die Signumfunktion SGN (X) gibt das Vorzeichen des Arguments in folgende Weise an:

  sgnx = + 1 für x > Ø
  sgnx = Ø für x = Ø
  sgnx = – 1 für x < Ø

- Mit Hilfe der Standardfunktion RND (X) können gleichverteilte Zufallszahlen zwischen 0 und 1 erzeugt werden. Dazu muß als Argument eine positive oder negative Zahl gewählt werden. Wählt man ein positives Argument, erhält man immer dieselbe Folge von Zahlen, während ein negatives Argument eine zufällige Folge von Zahlen erzeugt.

### 2.6 Arithmetische Ausdrücke und Zuordnungsanweisungen

*Arithmetische Ausdrücke* stellen Vorschriften zur Berechnung von Zahlenwerten dar. Sie bestehen aus Konstanten, Variablen und Standardfunktionen, die mit Hilfe von arithmetischen Operatoren verknüpft werden. Beim Aufbau von arithmetischen Ausdrücken sind einige Regeln zu beachten:

- Stehen mehrere arithmetische Operatoren verschiedenen Ranges in einer Anweisung, werden die Verknüpfungen entsprechend ihrer Rangfolge behandelt. Für die Rangordnung arithmetischer Operatoren gilt:

  Rang 1: ↑
  Rang 2: * und /
  Rang 3: + und –

- Kommen Standardfunktionen in arithmetischen Ausdrücken vor, wird ihnen der höchste Rang zugeordnet.
- Stehen mehrere gleichrangige arithmetische Operatoren in einem arithmetischen Ausdruck, werden sie in der Reihenfolge von links nach rechts behandelt.
- Der Multiplikationsoperator darf in arithmetischen BASIC-Ausdrücken nicht fortgelassen werden.
- Klammerausdrücke haben vor den arithmetischen Operationen Vorrang.
- Innere Klammerausdrücke haben Vorrang vor den äußeren Klammerausdrücken.
- Gleichrangige Klammerausdrücke werden von links nach rechts behandelt.
- Zwei arithmetische Operatoren dürfen nie unmittelbar aufeinander folgen.

Eine arithmetische *Zuordnungsanweisung* hat die allgemeine Form:

| n LET Variable = arithmetischer Ausdruck |
|---|

Bei einigen Heimcomputern kann das Schlüsselwort LET entfallen. Der Wert des rechtsstehenden arithmetischen Ausdrucks wird der linksstehenden Variablen zugeordnet. Wenn mehreren Variablen der gleiche Wert zugewiesen werden soll, sind bei einigen Heimcomputern Mehrfachzuweisungen möglich.

*Beispiel:*

| Mathematische Schreibweise | BASIC Schreibweise |
|---|---|
| $a = \frac{b \cdot c}{d \cdot e}$ | 1Ø LETA = B * C/D/E |

## 2.7 Steueranweisungen

In einem BASIC-Programm werden die Anweisungen in aufsteigender Reihenfolge der Anweisungsnummern geordnet und ausgeführt. Da die Anweisungsnummern bereits bei der Aufstellung des Programms vergeben werden, liegt die Bearbeitungsfolge der Anweisungen schon vor dem Programmablauf fest. Oft ist es jedoch wünschenswert, den Programmablauf in Abhängigkeit von berechneten oder eingegebenen Werten steuern zu können. Man unterscheidet dabei folgende Steueranweisungen:

- Sprunganweisungen
- Programmverzweigungsanweisungen
- Schleifenanweisungen
- Programmbeendungsanweisungen

### Unbedingte Sprunganweisungen

Die Ausführung der unbedingten Sprunganweisung GOTO führt zu einer unbedingten Programmverzweigung. Sie wird immer dann benutzt, wenn von einer Anweisung im Programm ohne jede einschränkende Bedingung zu einer anderen Anweisung im Programm gesprungen werden soll. Als Sprungziel wird die Anweisungsnummer dieser Anweisung angegeben. Nach erfolgtem Sprung wird das Programm *linear* weiter abgearbeitet bis gegebenenfalls eine andere Steueranweisung die Reihenfolge ändert.

Die unbedingte Sprunganweisung hat die Form:

| $n_1$ GOTO $n_2$ |
|---|

Hierbei ist:

- $n_1$ die Anweisungsnummer der unbedingten Sprunganweisung,
- GOTO das Schlüsselwort der unbedingten Sprunganweisung und
- $n_2$ das Sprungziel der unbedingten Sprunganweisung.

Die unbedingte Sprunganweisung bewirkt, daß das Programm mit der Anweisung der Anweisungsnummer $n_2$ fortgesetzt wird.

Beispiel für einen Vorwärtssprung:

```
.
.
5 GOTO 11
.
.
.
11 LET X = Y + Z
.
.
```

In diesem Ausschnitt eines Programmbeispieles, in dem einige Anweisungen durch Punkte symbolisch dargestellt wurden, wird gezeigt, daß mit Hilfe der unbedingten Sprunganweisung ein Programmteil übersprungen werden kann. In dem dargestellten *Vorwärtssprung* werden nach dem Befehl GOTO 11 drei Anweisungen übersprungen. Das Programm wird dann mit der Anweisung der Anweisungsnummer 11, d.h. mit der Anweisung LET X = Y + Z, fortgesetzt.

Das Sprungziel kann jedoch auch vor der unbedingten Sprunganweisung liegen. Man spricht dann von einem *Rücksprung.*

Beispiel für einen Rücksprung:

```
.
.
5 LET X = Y + Z
.
.
.
25 GOTO 5
.
```

Das Programmstück zwischen den beiden ausgeschriebenen Anweisungen wird immer wieder durchlaufen. Auf diese Weise kann man also eine *Programmschleife* bilden.

In diesem Beispiel wird die Programmschleife endlos lange durchlaufen. Um zu erreichen, daß eine Schleife nur endlich oft durchlaufen wird, muß in Abhängigkeit von einer Bedingung aus der Schleife herausgesprungen werden. Dies kann z. B. durch Programmverzweigungsanweisungen geschehen, die im folgenden besprochen wird.

**Programmverzweigungsanweisung**

Programmverzweigungsanweisungen bieten die Möglichkeit, das Programm in Abhängigkeit bestimmter Bedingungen zu verzweigen. Derartige Probleme werden in BASIC ähnlich wie in der Umgangssprache in Form einer *Wennanweisung* formuliert.

- Umgangssprache:

*Wenn* die Bedingung *B* erfüllt ist, *dann* soll die Operation *O* ausgeführt werden.

Die Programmverzweigungsanweisung hat in BASIC die Form:

$n_1$ IF $a_1 \oplus a_2$ THEN $n_2$

$n_1$ die Anweisungsnummer der Programmverzweigungsanweisung

IF zusammen mit THEN das Schlüsselwort der berechneten Sprunganweisung

$a_1, a_2$ die zu vergleichenden arithmetischen Ausdrücke

$\oplus$ das Symbol des Vergleichsoperators

$n_2$ die Anweisungsnummer der Anweisung, zu der verzweigt wird, wenn die Bedingung, ausgedrückt durch den Vergleich zweier arithmetischer Ausdrücke, erfüllt ist.

BASIC kennt sechs Vergleichsoperatoren

| Mathematisches Symbol | BASIC | Deutsche Sprechweise |
|---|---|---|
| $<$ | < | kleiner als |
| $\leqslant$ | <= | kleiner gleich |
| $=$ | = | gleich |
| $\geqslant$ | >= | größer gleich |
| $>$ | > | größer |
| $\neq$ | <> | ungleich |

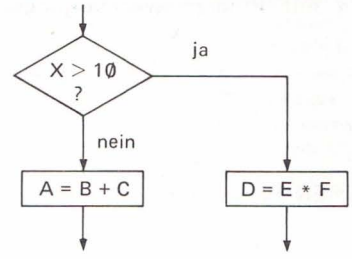
In dem Vergleichsausdruck $a_1 \oplus a_2$ muß das *allgemeine* Symbol $\oplus$ für den Vergleichsoperator durch einen dieser sechs *speziellen* Vergleichsausdrücke ersetzt werden.

Die Programmverzweigungsanweisung erlaubt somit eine Programmverzweigung in Abhängigkeit von dem Wahrheitswert eines Vergleichsausdrucks. Ist der Wahrheitswert des Vergleichsausdrucks $a_1 \oplus a_2$ wahr (die Bedingung ist erfüllt), dann wird zur Anweisung mit der angegebenen Anweisungsnummer $n_2$ verzweigt und daran anschließend, wie gewohnt, im Programm fortgefahren.

Beispiel für eine Programmverzweigungsanweisung:

```
20 IF X > 10 THEN 60
30 LET A = B + C
40  .
50  .
60 LET D = E * F
    :
    :
```

Im Programmablaufplan wird diese Programmverzweigungsanweisung wie folgt dargestellt:




**Schleifenanweisungen**

Eine Schleife ist eine Folge von mehrfach zu durchlaufenden Anweisungen. Eine Schleifenanweisung bewirkt, daß eine Folge von Anweisungen mehrfach durchlaufen wird. Programmschleifen lassen sich zwar schon mit Hilfe der geschilderten Sprung- und Wennanweisungen aufbauen. Eleganter ist jedoch die Verwendung einer speziellen Schleifenanweisung.

Die Schleifenanweisung besteht aus folgendem Anweisungspaar:

$n_1$ FOR v = $a_1$ TO $a_2$ STEP $a_3$
.
.
.
$n_2$ NEXT v

Die Schleife beginnt mit der FOR-Anweisung und endet mit der NEXT-Anweisung. Zwischen den Anweisungen FOR und NEXT sind die Anweisungen anzuordnen, die in der Schleife mehrfach durchlaufen werden sollen. Die Variable v ist eine sog. *Laufvariable.* Sie durchläuft von einem Anfangswert bis zu einem Endwert alle Werte mit einer vorgegebenen Schrittweite. Der *Anfangswert* (untere Grenze des Laufbereiches) wird durch den arithmetischen Aus-

druck $a_1$ festgelegt, der *Endwert* (obere Grenze des Laufbereiches) durch den arithmetischen Ausdruck $a_2$ und die *Schrittweite* durch den arithmetischen Ausdruck $a_3$.

Zur Kennzeichnung des Schleifenzusammenhanges muß die Variable v hinter FOR und NEXT stets die gleiche sein. $n_1$ und $n_2$ sind die Anweisungsnummern des Anweisungspaares der Schleifenanweisung. Der Schleifendurchlauf wird abgebrochen, wenn der Endwert von der Laufvariablen überschritten wird. Das Programm wird dann mit der nächsten Anweisung, die der Schleife folgt, fortgesetzt.

Die vorteilhaftere Schreibweise, die die Schleifenanweisung gegenüber einer Konstruktion aus Sprung- und Wennanweisungen bietet, kann anhand des Beispiels 1 aus dem Beitrag „Grafische Darstellung von Programmabläufen" demonstriert werden:

- Programmschleife mit Programmverzweigungsanweisung

```
 5 LET S = Ø
1Ø LET I = 2
15 LET S = S + I
2Ø LET I = I + 2
25 IF I < = 2Ø THEN 15
3Ø ...
 :
```

- Programmschleife mit Schleifenanweisung

```
 5 LET S = Ø
1Ø FOR I = 2 TO 2Ø STEP 2
15 LET S = S + I
2Ø NEXT I
25 ...
 :
```

Bei der Verwendung der Schleifenanweisung sind folgende Regeln zu beachten:

- Die Anzahl der BASIC-Anweisungen zwischen FOR- und NEXT-Anweisung ist unbegrenzt.
- Wenn der Wert des arithmetischen Ausdrucks $a_3$ positiv ist, muß der Wert von $a_2$ größer als der von $a_1$ sein. Ist dies nicht der Fall, wird die Schleife nicht durchlaufen und das Programm fährt mit der auf NEXT folgenden Anweisung fort.

  Wenn der Wert des arithmetischen Ausdrucks $a_3$ negativ ist, muß hingegen der Wert von $a_1$ größer als der von $a_2$ sein.
- Der arithmetische Ausdruck $a_3$, der die Schrittweite angibt, kann entfallen, wenn $a_3 = 1$ ist. Dies vereinfacht die Formulierung der Programmschleife.
- Der Wert der Laufvariablen v, sowie die Werte der arithmetischen Ausdrücke $a_1$, $a_2$ und $a_3$ dürfen nicht durch Anweisungen innerhalb des Laufbereiches verändert werden.

  Innerhalb einer Schleife ist es z. B. verboten, die Laufvariable v durch eine Anweisung wie z. B. V = V + 1 zu verändern.
- Aus dem Schleifenbereich darf zwar herausgesprungen werden, aber nicht hinein. Die Laufvariable hat außerhalb der Schleifenanweisung den Wert, den sie unmittelbar vor Ausführung der Sprunganweisung hatte. Dieser Wert ist dann außerhalb der Schleife verfügbar.
- Es können mehrere Schleifenanweisungen geschachtelt werden. Die innere Schleife muß dabei vollständig in der äußeren Schleife liegen. Die maximale Anzahl der ineinander geschachtelten Schleifen ist abhängig von der jeweiligen DVA und ist aus den Herstellerhandbüchern zu entnehmen.

**Programmbeendungsanweisungen**

Jedes BASIC-Programm muß mit einer *END-Anweisung* abgeschlossen werden. Sie muß die höchste vergebbare Anweisungsnummer besitzen.

Die Endanweisung hat die Form:

| n END |
|---|

Die *Stopanweisung* bewirkt einen Sprung zum Programmende. Sie hat die Form:

| n STOP |
|---|

## 2.8 Eingabeanweisungen

Eingabeanweisungen dienen dazu, Programme mit den erforderlichen Daten zu versorgen.

- Eingabe mit Hilfe der READ-DATA-Anweisung

Daten werden mit der READ-DATA-Anweisungen folgendermaßen eingegeben:

| $n_1$ READ $v_1, v_2, \ldots, v_n$ |
|---|
| . |
| . |
| . |
| $n_2$ DATA $K_1, K_2, \ldots, K_n$ |

Mit der READ-DATA-Anweisung können beliebig vielen Variablen Werte zugewiesen werden. Diese

Variablen werden, durch Kommata getrennt, in einer „Variablenliste" hinter dem Schlüsselwort READ aufgelistet. Die Reihenfolge der Variablen kann beliebig sein. In der Variablenliste können auch indizierte Variablen (Felder) stehen. Die Zahlenwerte der Variablen werden, durch Kommata getrennt, in einer „Werteliste" hinter dem Schlüsselwort DATA aufgelistet und zwar in der Reihenfolge, in der die zugehörigen Variablen in der Variablenliste der READ-Anweisung aufgelistet sind. Die Variablen der Variablenliste müssen nicht in *einer* READ-Anweisung zusammengefaßt sein. Die Daten für die Variablen der Variablenliste müssen nicht in *einer* DATA-Anweisung zusammengefaßt sein.

Beispiel einer Eingabeanweisung mit Hilfe von READ-DATA:

| Eingabeanweisung | Erläuterung |
|---|---|
| 1Ø READ A, B<br>.<br>.<br>.<br>5Ø DATA 15, 18<br>.<br>.<br>. | Der Variablen A wird der Wert 15 und der Variablen B der Wert 18 zugewiesen. |

- Eingabe mit Hilfe der INPUT-Anweisung

Die INPUT-Anweisung hat folgende allgemeine Form:

| n INPUT $v_1, v_2, \ldots, v_n$ |
|---|

Mit der INPUT-Anweisung können Daten während des Programmlaufes eingegeben werden. Die Variablen, für die Werte eingegeben werden sollen, werden in einer „Variablenliste" hinter dem Schlüsselwort INPUT, durch Kommata getrennt, aufgelistet. In der Variablenliste können auch indizierte Variablen (Felder) stehen. Die Reihenfolge der Variablen kann beliebig sein. Bei der Ausführung der INPUT-Anweisung wird ein Fragezeichen auf der Benutzerstation ausgegeben. Das Programm erwartet daraufhin die Eingabe der Zahlenwerte für die Variablen der Variablenliste der INPUT-Anweisung. Die Zahlenwerte für die einzelnen Variablen werden durch Kommata voneinander getrennt.

[1]) Die Unterteilung ist z. T. herstellerabhängig. Beim Heimcomputer PET wird z. B. die Ausgabezeile wegen des kleinen Bildschirms in 4 Felder zu je 10 Druckstellen unterteilt.

Beispiel einer INPUT-Eingabeanweisung:

| INPUT-Anweisung | Erläuterung |
|---|---|
| .<br>.<br>.<br>4Ø INPUT A, B<br>5Ø LET X = A + B<br>.<br>.<br>.<br>8Ø END<br>? 4.97, Ø.5 | Während des Rechenlaufes eines Programms möge die DVA auf die INPUT-Anweisung (4Ø INPUT A, B) stoßen. Sie wird gelesen und als Eingabeanweisung interpretiert.<br>Bei der Ausführung der INPUT-Anweisung wird von der DVA auf dem Sichtgerät ein Fragezeichen ausgedruckt. Das Programm erwartet nun die Eingabe von zwei Zahlenwerten für die Eingabevariablen A und B.<br>Der Benutzer gibt über die Tastatur des Heimcomputers zunächst für die Variable A einen Wert ein, z. B. 4,97, trennt diesen Wert gegenüber den folgenden durch ein Komma und gibt anschließend für die Variable B den Wert – 0,5 ein.<br>Nach Abschluß der Eingabe (Drücken der RETURN-Taste o.ä.) wird zur nächsten Anweisung des Programms (5Ø LET X = A + B) übergegangen. |

## 2.9 Ausgabeanweisungen

Ausgabeanweisungen dienen dazu, Daten programmgesteuert auszugeben. Daten werden mit Hilfe der PRINT-Anweisung folgendermaßen ausgegeben:

| n PRINT $a_1, a_2, \ldots, a_n$ |
|---|
| bzw. n PRINT $a_1; a_2; \ldots; a_n$ |

Mit Hilfe des Schlüsselwortes PRINT wird der DVA mitgeteilt, daß Daten ausgegeben werden sollen. Die arithmetischen Ausdrücke $a_1, \ldots, a_n$, deren Werte ausgegeben werden sollen, werden in einer Liste hinter dem Schlüsselwort PRINT aufgelistet. Die Liste kann eine beliebige Anzahl von arithmetischen Ausdrücken enthalten. Ihre Reihenfolge ist beliebig. Sie werden durch Listentrennzeichen (Kommas, Semikolons) voneinander getrennt.

- Das Standard-Spaltenformat
  - Die Trennung der Liste der arithmetischen Ausdrücke durch Kommas bewirkt, daß die Ausgabezeile standardmäßig z.B. in fünf Felder zu je fünfzehn Druckstellen unterteilt wird[1]).
  - Für jeden arithmetischen Ausdruck, dessen Wert ausgegeben werden soll, ist ein Feld vorgesehen (siehe **Bild 2.2**).

– Die einzelnen Ausgabewerte werden der Reihe nach linksbündig in die jeweiligen Felder gedruckt.
– Wenn mehr als fünf Werte ausgegeben werden sollen, wird automatisch zu einer neuen Zeile übergegangen.
– Falls ein zu druckender Wert mehr als 15 Druckstellen benötigt, werden dazwischenliegende Feldgrenzen nicht berücksichtigt.

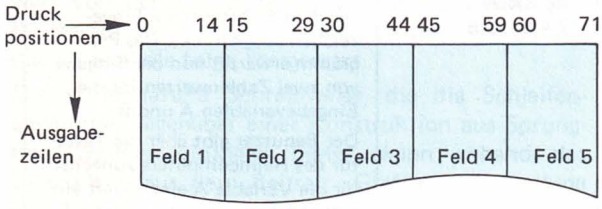



**Bild 2.2** Einteilung der Ausgabezeilen in Felder

Beispiel für die Ausgabe von Variablenwerten im Standard-Spaltenformat:

Den Variablen A, B, C, D, E, F, G, H, I wurden in einer Rechnung die ganzen positiven Zahlen von 1 bis 9 zugewiesen (**Bild 2.3**).

– Für jeden arithmetischen Ausdruck, dessen Wert ausgegeben werden soll, ist ein Feld vorgesehen, dessen Feldlänge von der Anzahl der auszugebenden Ziffern abhängt.
– Die einzelnen Ausgabewerte werden der Reihe nach linksbündig in die jeweiligen Felder gedruckt.
– Wenn mehr Werte ausgegeben werden sollen, als Felder in der Zeile vorhanden sind, wird automatisch zu einer neuen Zeile übergegangen.

*Beispiel:*

Es sollen mit Hilfe des *variablen* Spaltenformats folgende Werte in 3 Zeilen ausgegeben werden:

Zeile 1: 1; 2; 3
Zeile 2: 12; 123; 1234
Zeile 3: 12345; 123456; 1234567

Die Befehlsfolge lautet dazu:

```
50 PRINT 1; 2; 3
60 PRINT 12; 123; 1234
70 PRINT 12345; 123456; 1234567
```

Das Druckbild sieht dann wie in **Bild 2.4** aus.

| PRINT-Anweisung | 70 PRINT A, B, C, D, E, F, G, H, I |
|---|---|
| Druckbild der Ausgabe | Spalten-position 0 14 15 29 30 44 45 59 60 71 |

**Bild 2.3**

● Das variable Spaltenformat

– Trennt man die arithmetischen Ausdrücke in der Liste der arithmetischen Ausdrücke durch ein Semikolon, so werden die Werte in einem variablen Spaltenformat ausgedruckt.

Eine häufig benutzte Formatierung ist folgender Tabelle zu entnehmen:

| Anzahl der Ziffern in der Zahl | Anzahl der Druckstellen je Feld | Anzahl[1]) der Felder |
|---|---|---|
| 1 | 3 | 24 |
| 2, 3, 4 | 6 | 12 |
| 5, 6, 7 | 9 | 8 |
| 8, 9, 10 | 12 | 6 |
| ≥ 11 | 15 | 5 |

[1]) Herstellerabhängig.

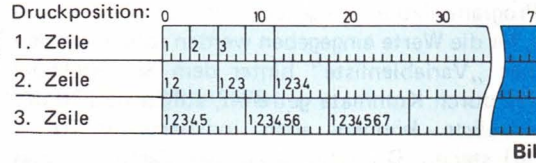



**Bild 2.4**

Wie dieses Beispiel deutlich zeigt, ergibt die Verwendung des variablen Spaltenformats keine tabellarische Anordnung der Werte.

● Das Tabellenformat

Häufig steht der Programmierer vor dem Problem, errechnete Werte in Tabellenform ausgeben zu müssen. BASIC bietet eine andere Möglichkeit, Berechnungsergebnisse tabellarisch in jedem gewünschten Format auszudrucken. Den tabellarisch auszudruckenden arithmetischen Ausdrücken der PRINT-Anweisung müssen zu diesem Zweck entsprechende *Tabulatorfunktionen* (TAB) vorangestellt werden.

– Die allgemeine Form der Ausgabeanweisung für Tabellen ist:

| | |
|---|---|
| | n PRINT TAB ($a_1$), $a_2$, TAB ($a_3$), $a_4$, . . . , TAB ($a_{n-1}$); $a_n$ |
| bzw. | n PRINT TAB ($a_1$); $a_2$; TAB ($a_3$); $a_4$; . . . ; TAB ($a_{n-1}$); $a_n$ |

– Mit Hilfe des Schlüsselwortes TAB wird der DVA mitgeteilt, daß eine tabellarische Ausgabe folgen soll.
– Der Wert des arithmetischen Ausdrucks $a_{n-1}$ im Argument der Tabulatorfunktion (runde Klammer) gibt die Spaltenposition in der jeweiligen Druckzeile an, in der mit dem Drucken des gewünschten Wertes begonnen werden soll.
– Der gewünschte Wert ergibt sich aus dem arithmetischen Ausdruck $a_n$, der auf die Tabulatorfunktion folgt.
– Als Listentrennzeichen sind sowohl Kommas als auch Semikolons möglich.

  Das Komma reserviert für den jeweiligen Ausgabewert ein Feld von 15 Spalten.

  Das Semikolon reserviert für den jeweiligen Ausgabewert ein Feld, dessen Größe von der Anzahl der Ziffern des Wertes abhängt.
– In der Ausgabeanweisung für Tabellen dürfen auch arithmetische Ausdrücke ohne Tabulatorfunktion auftreten.
– Die Tabulatorfunktion wird unwirksam, wenn die im Argument der Tabulatorfunktion angegebene Druckposition kleiner als die augenblickliche, schon erreichte Druckposition ist.

  Ist das Argument größer als 71, wird zu einer neuen Zeile übergegangen. Ist das Argument negativ, bleibt der Schreibkopf am linken Zeilenrand stehen.

- Der Zeilenvorschub

– Jede PRINT-Anweisung bewirkt, daß zum Drucken der Werte eine neue Zeile eingenommen wird.
– Wenn die auf das Schlüsselwort PRINT folgende Liste der arithmetischen Ausdrücke mit einem Listentrennzeichen (Komma, Semikolon) abschließt, wird eine *folgende* PRINT-Anweisung in der bisherigen Zeile weiterdrucken.
– Eine Leerzeile wird durch eine PRINT-Anweisung erzeugt, die keine Liste mit arithmetischen Ausdrücken enthält.

Die allgemeine Form zum Drucken einer Leerzeile ist somit:

| |
|---|
| n PRINT |

- Ausgabe von kommentierenden Texten

Kommentierende Texte können mit Hilfe der PRINT-Anweisung ausgegeben werden.

Die allgemeine Form zur Ausgabe von festen Texten ist:

| |
|---|
| n PRINT "Text" |

– Der Text darf aus beliebigen Zeichen des BASIC-Zeichenvorrats bestehen und ist in Anführungszeichen (" ") zu setzen.
– Leerstellen innerhalb eines durch Anführungszeichen gekennzeichneten Textes werden als „leere" Druckstellen gewertet, d. h. es wird kein BASIC-Zeichen ausgedruckt.
– Es können mehrere kommentierende Texte in einer Zeile ausgegeben werden, indem man die Texte durch die Listentrennzeichen (Komma, Semikolon) voneinander trennt.

Die allgemeine Form zur Ausgabe von n festen Texten ist:

| | |
|---|---|
| | n PRINT "Text 1", "Text 2", . . . , "Text n" |
| bzw. | n PRINT "Text 1"; "Text 2"; . . . ; "Text n" |

Verwendet man als Listentrennzeichen ein Komma, so werden die Druckzeilen in 5 Felder mit je 15 Druckstellen aufgeteilt, in die die Texte der Reihe nach linksbündig ausgedruckt werden.

Nach Verwendung des letzten Feldes der Druckzeilen wird zum ersten Feld der nächsten Zeile übergegangen.

Verwendet man als Listentrennzeichen ein Semikolon, so werden die einzelnen Texte ohne Zwischenraum aneinandergereiht.

- Ausgabe von kommentierenden Texten und Daten

Texte und arithmetische Ausdrücke können in einer PRINT-Anweisung vermischt auftreten.

Die allgemeine Form zur Ausgabe von Texten und Daten ist:

| | |
|---|---|
| | n PRINT "Text 1", $a_1$, . . . , "Text n", $a_n$ |
| bzw. | n PRINT "Text 1"; $a_1$; . . . ; "Text n"; $a_n$ |

Verwendet man als Listentrennzeichen ein Komma, so wird die Druckzeile in bekannter Form in 5 Felder mit je 15 Druckstellen aufgeteilt. Verwendet man als Listentrennzeichen ein Semikolon, so wird zwischen kommentierendem Text und einem darauf folgenden Wert nur eine Druckstelle als Zwischenraum frei gehalten.

*Beispiel:*

| Druckanweisung | Ausdruck (Druckzeile) |
|---|---|
| 100 PRINT "X="; X<br>Der Wert von X sei 10,5 | X = 10,5 |

**Kommentare im Programm**

Kommentare sind zusätzliche Bemerkungen oder Erklärungen, die dem Programmierer helfen, das Programm übersichtlich zu gestalten.

Kommentare können an beliebigen Stellen im Programm stehen.

Die Kommentarvereinbarung hat folgende allgemeine Form:

```
n REM Text
```

n Anweisungsnummer

REM Schlüsselwort für die Kommentarvereinbarung

Text Der kommentierende Text darf beliebige Zeichen aus dem Fernschreiber-Zeichenvorrat enthalten. Die Zahl der Zeichen darf die maximale Zeichenzahl einer Programmzeile, z. B. 80 Zeichen, nicht überschreiten. Ist der Text länger, muß eine neue Kommentarvereinbarung in einer neuen Programmzeile begonnen werden.

Bei der Kommentarvereinbarung handelt es sich um keine Anweisung, die dem Fortgang des Programms dient.

Daher wird sie bei der Übersetzung des Programms nicht berücksichtigt.

Der Kommentar erscheint lediglich bei Programmauflistungen.

## 3 Beispiel: Zinseszins- und Rentenrechnung

*Aufgabenstellung*

Im Geschäfts- und Privatleben wird man vielfach mit dem Problem konfrontiert, daß ein gewisses Kapital zinsbringend anzulegen ist, welches man in festen Raten für eine gewisse Zeit spart (z. B. nach dem 624,– DM-Gesetz) oder von einem Kapital gleichbleibende Abhebungen vornimmt. Der Sparer wird im allgemeinen wissen wollen, auf welches Endkapital sein Anfangskapital nach einer gewissen Zeit bei einem vorgegebenen Zinssatz wächst bzw. fällt. Für diese Aufgabe ist ein möglichst universelles Programm zu erstellen.

*Problemformulierung*

Mit Hilfe der *Zinseszinsrechnung* kann ermittelt werden, auf welches Endkapital $K_n$ ein Anfangskapital $K$ innerhalb von $n$ Jahren bei einem Zinssatz von $p$ % anwächst, wenn die am Ende des Jahres fälligen Zinsen dem Kapital zugefügt und weiterhin mitverzinst werden.

Die Zinseszinsformel ergibt sich nach *Leibnitz* zu

$$K_n = K \cdot \left(1 + \frac{p}{100}\right)^n = K \cdot q^n$$

Der Zinsfaktor $(1 + \frac{p}{100})$ wird vielfach auch mit $q$ bezeichnet. **Fügt** man zum vorhandenen Kapital $K$ außerdem noch jährlich am Jahresanfang eine Sparrate $R$ **hinzu** oder **hebt** man jährlich einen gleichbleibenden Betrag $R$ vom Kapital **ab**, so ist bei $n$ Jahren und $p$ % folgende **Rentenformel** anzuwenden:

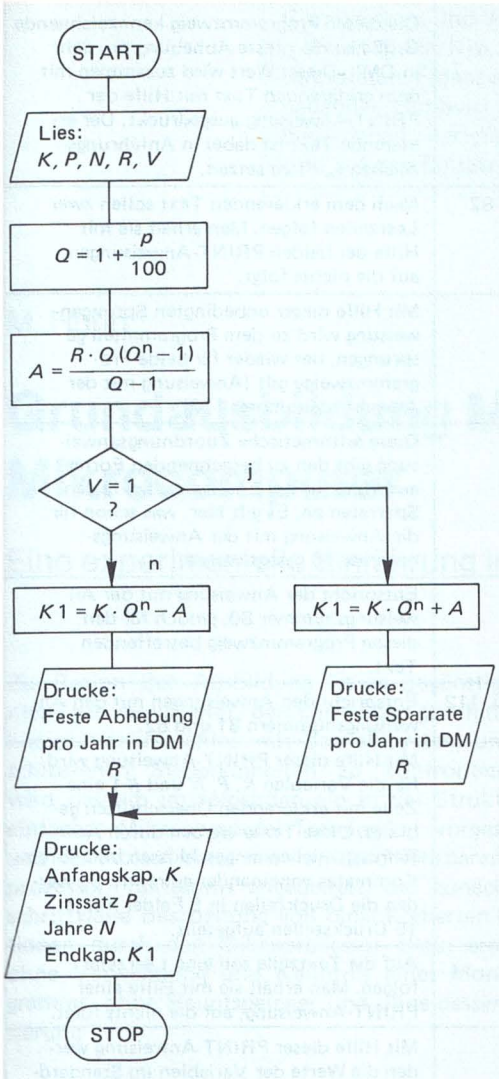
$$K_n = K \cdot q^n \pm \frac{R \cdot q\,(q^n - 1)}{q - 1} \quad \Bigg| \quad \text{mit } q = \left(1 + \frac{p}{100}\right)$$

Die Addition findet Anwendung, wenn jährlich Raten hinzugefügt werden, die Subtraktion hingegen, wenn jährlich gleichbleibende Abhebungen vorgenommen werden.

Für den Rechenlauf sind beispielhaft folgende Aufgaben zu lösen:

1. Eine Großmutter gibt zur Taufe ihres Enkels DM 1000,– auf ein Sparbuch und jährlich 100,– DM zum Geburtstag. Der Zinssatz beträgt 4 %. Wie groß ist das Endkapital am 20. Geburtstag?
2. Ein Geschäftsmann setzt sich mit 500 000,– DM zur Ruhe. Er legt dieses Geld langfristig zu 6 % an. Zum Lebensunterhalt benötigt er 60 000,– DM jährlich. Wie groß ist sein Vermögen nach 10 Jahren? Hätte er, ohne Schulden zu machen, auch 65 000,– DM abheben können?

**Programmablaufplan**




*Programm- und Ergebnisausdruck*

```
10 REMZINSESZINS UND RENTENRECHNUNG
20 REM
30 INPUT K,P,N,R,V
40 LET Q=1+P/100
50 LET A=(R*Q*(Q↑N-1))/(Q-1)
60 IF V=1 THEN 100
70 K1=K*Q↑N-A
80 PRINT "FESTE ABHEBUNG PRO JAHR IN DM";R
81 PRINT
82 PRINT
90 GOTO 120
100 K1=K*Q↑N+A
110 PRINT "FESTE SPARRATE PRO JAHR IN DM "R
111 PRINT
112 PRINT
120 PRINT  ANFANGSKAP. K ","ZINSSATZ P","JAHRE N","ENDKAP. K1"
130 PRINT
140 PRINT K,P,N,K1
150 END
```

```
FESTE SPARRATE PRO JAHR IN DM  100

ANFANGSKAP. K   ZINSSATZ P      JAHRE N          ENDKAP. K1

 1000            4               20               5288.043315


FESTE ABHEBUNG PRO JAHR IN DM 60000

ANFANGSKAP. K   ZINSSATZ P      JAHRE N          ENDKAP. K1

 500000          6               10               57125.28994


FESTE ABHEBUNG PRO JAHR IN DM 65000

ANFANGSKAP. K   ZINSSATZ P      JAHRE N          ENDKAP. K1

 500000          6               10               -12732.92326
```

*Erläuterungen zum Programmablaufplan*

Im Programmablaufplan wird dargestellt, wie die mathematische Formel schrittweise aufgebaut wird. Je nachdem, ob im Programm feste Sparraten oder feste Abhebungen behandelt werden sollen, wird das Programm in Abhängigkeit vom Wert der Variablen $V$ verzweigt. Ist $V = 1$, werden feste Sparraten behandelt. Für alle anderen Werte für $V$ werden feste Abhebungen behandelt.

*Erläuterungen zum Programm*

| Anw. Nr. | Erläuterung |
|---|---|
| 10 | Die Programmüberschrift wird mit Hilfe einer Kommentarvereinbarung formuliert. |
| 20 | Um die Programmüberschrift vom eigentlichen Programm abzuheben, wurde nach der Überschrift eine Kommentarvereinbarung ohne Kommentar eingefügt. |
| 30 | Eingabe von Daten mit Hilfe der INPUT-Anweisung. Auf das Schlüsselwort INPUT folgt die Variablenliste mit der Variablen *K* für das Anfangskapital, *P* für den Zinssatz, *N* für die Zahl der Jahre, *R* für die jährliche Sparrate bzw. für die jährliche Abhebung und *V* für die Variable, mit der festgelegt wird, ob die Variable *R* eine jährliche Sparrate oder eine jährliche Abhebung darstellen soll. Die Variablen werden durch Kommata voneinander getrennt. |
| 40 | Diese arithmetische Zuordnungsanweisung gibt den zu berechnenden Formelausdruck für den Zinsfaktor *q* an. Der arithmetische Ausdruck wird gemäß der Regel „Punktrechnung vor Strichrechnung" abgearbeitet, d.h. es wird zunächst *P* durch 100 dividiert und dann zu diesem Ergebnis 1 addiert. |
| 50 | Diese arithmetische Zuordnungsanweisung gibt den zu berechnenden Formelausdruck für die verzinste jährliche Sparrate bzw. für die verzinste jährliche Abhebung an. Für die Potenzbildung wird das Sonderzeichen ↑ benutzt. |
| 60 | Diese Programmverzweigungsanweisung verzweigt zur Anweisungsnummer 100, wenn für die Variable *V* = 1 eingegeben wird. Ansonsten wird zum Befehl mit der nächsten Anweisungsnummer, d.h. zu Anweisungsnummer 70 verzweigt.<br>In den Befehlen mit den Anweisungsnummern 70 bis 90 wird der Programmteil mit den festen Abhebungen behandelt, in den Befehlen mit den Anweisungsnummern 100 bis 112 der Programmteil mit den festen Sparraten. |
| 70 | Diese arithmetische Zuordnungsanweisung gibt den zu berechnenden Formelausdruck für das Endkapital bei Abhebungen an. Der in der Formel vorgesehene Variablenname $K_n$ kann in BASIC allerdings nicht vergeben werden, da in BASIC das erste Zeichen ein Buchstabe und das zweite Zeichen eine Ziffer sein muß. Weiterhin wurde hier auf das Schlüsselwort LET verzichtet, da die benutzte DVA dies zuläßt. |

| Anw. Nr. | Erläuterung |
|---|---|
| 80 | Die diesen Programmzweig kennzeichnende Größe ist die „feste Abhebung pro Jahr in DM". Dieser Wert wird zusammen mit dem erklärenden Text mit Hilfe der PRINT-Anweisung ausgedruckt. Der erklärende Text ist dabei in Anführungszeichen („ ") zu setzen. |
| 81 u. 82 | Nach dem erklärenden Text sollen zwei Leerzeilen folgen. Man erhält sie mit Hilfe der beiden PRINT-Anweisungen, auf die nichts folgt. |
| 90 | Mit Hilfe dieser unbedingten Sprunganweisung wird zu dem Programmteil gesprungen, der wieder für beide Programmzweige gilt (Anweisung mit der Anweisungsnummer 120). |
| 100 | Diese arithmetische Zuordnungsanweisung gibt den zu berechnenden Formelausdruck für das Endkapital bei festen Sparraten an. Es gilt hier, was schon für die Anweisung mit der Anweisungsnummer 70 gesagt wurde. |
| 110 | Entspricht der Anweisung mit der Anweisungsnummer 80, jedoch für den diesen Programmzweig betreffenden Text. |
| 111 u. 112 | Entspricht den Anweisungen mit den Anweisungsnummern 81 und 82. |
| 120 | Mit Hilfe dieser PRINT-Anweisung wird für die Variablen *K, P, N* und *K*1 eine Zeile mit erklärenden Überschriften gebildet. Diese Texte werden durch Anführungszeichen eingeschlossen und durch Kommatas voneinander getrennt. So werden die Druckzeilen in 5 Felder mit je 15 Druckstellen aufgeteilt. |
| 130 | Auf die Textzeile soll eine Leerzeile folgen. Man erhält sie mit Hilfe einer PRINT-Anweisung, auf die nichts folgt. |
| 140 | Mit Hilfe dieser PRINT-Anweisung werden die Werte der Variablen im Standard-Spaltenformat ausgegeben. |
| 150 | Diese Anweisung gibt das Programmende an. Sie besitzt die höchste vergebbare Anweisungsnummer. |

## Literatur


[1] *Kemeny, J. G.* und *Kurtz, T. E.:* BASIC Programming – John Wiley & Sons, New York 1972.

[2] *American National Standards Institute* (ANSI): Proposed American National Standard for Minimal BASIC X3J2/76–35, Dec. 1976.

[3] *European Computer Manufacturers Association* (ECMA): Standard ECMA – Minimal BASIC – Final Draft ECMA/TC21/76/44 Dec. 1976.

Für Schulen und zum Selbststudium gibt es eine Vielzahl von $\mu$C-Lernsystemen. Grundaufbau und Ausstattung hängen wesentlich von der Zielgruppe ab, manchmal von ihrem Niveau (z.B. Grundschüler oder Fachstudenten), ein andermal vom vermuteten Finanzvolumen (vielleicht Taschengeldempfänger einerseits oder wohlhabender Hobbyist andererseits). Im folgenden Aufsatz wird ein Lerncomputer beschrieben, der nach didaktischen Gesichtspunkten entwickelt wurde und deshalb auf den Einsatz im Informatik-Unterricht und auf das Selbststudium abzielt.

M. Hund

# Grundausbildung Mikroprozessor und Mikrocomputer

## Eine experimentelle Einführung in Struktur und Arbeitsweise

Zu Beginn der Ausbildung ist es ungünstig (wenn nicht gar verwirrend), den Lernenden gleich mit einer kompletten Maschine mit **Betriebssystem** und angeschlossenen Steuereinheiten zu konfrontieren. Es wird deshalb im folgenden ein in der Struktur sehr einfacher Mikrocomputer-Training-Set vorgeschlagen und der Ablauf eines mit ihm durchführbaren Mikroprozessor-Praktikums angedeutet, der zunächst eine schrittweise Bearbeitung von binär-kodierten Instruktionen durch den Mikroprozessor allein ermöglicht ohne Verwendung und Kenntnis eines **Monitorprogramms**, ohne Hauptspeicher und ohne dessen Adressierung.

### 1 Instruktionseingabe über Datenbus in den Mikroprozessor

Der Mikroprozessor ist nur ein Teil eines Computers und enthält **Steuerwerk, Rechenwerk** und oft mehrere **Arbeitsregister**. Die Kommunikation mit seiner Umwelt geschieht über zahlreiche (z.B. 40) Anschlüsse, die sich in **Datenbus** (z.B. 8), **Steuerbus** und **Adreßbus** (z.B. 16) aufteilen lassen.

Im Mikrocomputer ist der Mikroprozessor (z.B. Typ: SAB 8080 A) verknüpft mit verschiedenen Speichern und Ein-/Ausgabemöglichkeiten. Ein spezieller Speicher enthält bereits festabgespeicherte Programmteile zur Unterstützung des Bedieners, zur Automatisierung einer Programmeingabe in den Mikrocomputer, zur Umwandlung und Übertragung von Informationen z.B. auf oder von Band und zu anderen Zwecken mehr. Die verschiedenen Funktionen dieses Monitorprogramms können im vorgeschlagenen Mikrocomputer-Training-Set über spezielle Programmschalter angewählt werden.

Sind alle Programmschalter auf Null gesetzt, wird das Monitorprogramm nicht angewählt, und das Steuerwerk im Mikroprozessor erhält seine Anweisungen

**Betriessystem:** Programme, die den „Betrieb" einer DV-Maschine ermöglichen (vom Hersteller fest eingespeichert).

**Monitor:** Teile eines Betriebssystems, die das Einspeichern von Daten und das Anzeigen von Adressen und darin enthaltenen Daten ermöglichen.

**Steuerwerk, Rechenwerk:** Teile einer Computer-Zentraleinheit, die den gesamten Arbeitsablauf steuern und die durch das Programm gewünschten Verrechnungen ausführen.

**Arbeitsregister:** Spezielle Speicherstellen, die Rechenwerte und Zwischenergebnisse aufnehmen können.

**Datenbus, Steuerbus, Adreßbus:** Gesamtheit aller Leitungen (Bus) für die Übertragung von Daten, Steuersignalen und Speicheradressen.

von Anwender in Form von einzelnen **Instruktionen** die in zeitlicher Abfolge über den Datenbus (**Datenbreite 8 Bit = 1 Byte**) zugeführt werden. Eine Instruktion kann dabei 1, 2 oder 3 Byte lang sein und gibt an, was und womit etwas getan werden soll.

Von den Steueranschlüssen des Mikroprozessors werden hier der **Reset-** und der **Ready-Eingang** benötigt, und ein besonderer Eingabekanal (8 Bit) ist auf den Datenbus geschaltet.

Wird Reset-Signal gegeben, so beginnt der Mikroprozessor mit einem Instruktionsabruf und wartet auf die Eingabe der ersten Instruktion (oder des ersten Instruktionsteils) über den Datenbus. Die einzelnen Bits des Eingabekanals können von Hand gesetzt werden und Kontrolldioden leuchten entsprechend auf.

Erst mit Ready-Signal (hier durch Druck auf eine Taste) übernimmt der Mikroprozessor die auf dem Datenbus liegende Instruktion in sein Befehlsregister. Der Inhalt des **Eingabekanals** wird dabei automatisch gelöscht.

Im Mikroprozessor wird anschließend der Inhalt des Befehlsregisters decodiert. Ist die Instruktion bereits komplett, wird sie unmittelbar ausgeführt. Anschließend erwartet der Mikroprozessor auf dem Datenbus die nächste Instruktion oder den nächsten Instruktionsteil.

Am Beispiel der aus zwei Byte bestehenden Instruktion „Lade Konstante in den Akkumulator" sei die Prozedur noch einmal kurz beschrieben. Nach Setzen des Eingabekanals (8 Bit) mit dem ersten Teil der Instruktion wird Ready-Signal gegeben. Die Instruktion gelangt ins Befehlsregister, wird decodiert, und der Prozessor wartet auf den zweiten Teil der Instruktion, auf die Konstante, die wiederum auf den Eingabekanal geschrieben wird. Nach erneutem Ready-Signal übernimmt der Mikroprozessor den auf dem Datenbus liegenden zweiten Teil der Instruktion als Konstante in den **Akkumulator** und wartet dann auf die nächste Instruktion.

## 2 Anwahl von Ausgabe- und Eingabekanälen

Soll der Inhalt des Akkumulators überprüft werden, kann eine besondere Ausgabeinstruktion eingegeben werden. Auch diese besteht aus zwei Byte. Nach Übernahme des zweiten Bytes (in dem später eine Kennung zur Anwahl verschiedener Ausgabekanäle stehen kann) gibt der Mikroprozessor den Inhalt des Akkumulators über den Datenbus auf einen angeschlossenen Ausgabekanal (8 Bit), der die Information speichert und mit Kontrolldioden optisch anzeigt. Die bisher angesprochenen Baugruppen des Computers sind in **Bild 1** zu sehen.

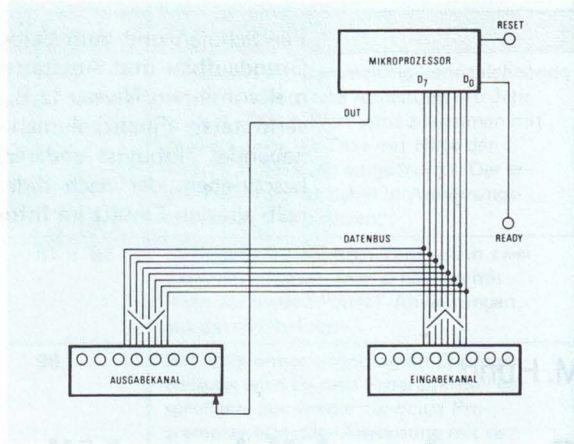



**Bild 1** Anwahl eines Ausgabekanals

Der vorgeschlagene Mikrocomputer besitzt neben den eben benutzten Eingabe- und Ausgabekanälen zu je 8 Bit noch je zwei weitere Eingabe- und Ausgabekanäle zu je einem Bit, die ebenfalls mit Leuchtdioden und Buchsen angeordnet sind. Die verschiedenen Kanäle werden durch das zweite Byte der Ausgabe- bzw. einer neuen Eingabeinstruktion unterschieden. Später werden zur Datenübertragung auf oder von Band, zur hexadezimalen Anzeige von Daten und Adressen und zur Anwahl der Tastatur noch weitere Ausgabe- und Eingabekanäle benötigt und beschrieben.

## 3. Arbeitsregister und Rechenwerk

Der eingebaute Mikroprozessor enthält neben Akkumulator, Befehlsregister und Decodierer noch weitere Funktionsgruppen, deren Existenz und Eigenschaft

**Instruktion:** Fachausdruck für Programmbefehl.

**Datenbreite:** Gibt an, wieviele Bit gleichzeitig (parallel) verarbeitet werden (Wortbreite). Wichtig: 4, 8 oder 16 Bit.

**Reset:** Rücksetzen. Einschalten eines definierten Anfangszustands.

**Ready:** Fertig-Meldung.

**Kanal:** Einzelleitung bei serieller Übertragung von Daten oder Leitungsbündel (Bus) bei paralleler Übertragung (entsprechend der Wortbreite).

**Akkumulator:** Über dieses wichtigste Register eines Computers werden die meisten Operationen ausgeführt.

durch den experimentellen Umgang mit dem Prozessor allein erkannt werden kann. Mit der Instruktion „Lade Konstante in Arbeitsregister" und einer universellen Instruktion zum Datentransport zwischen insgesamt sechs Arbeitsregistern wird deren Speicherverhalten nachgewiesen. Mit arithmetischen und logischen Instruktionen wird anschließend das Rechenwerk, die **ALU ( Arithmetic Logic Unit)**, mit seinen möglichen Verknüpfungen zwischen Datenbus, Arbeitsregister und Akkumulator aufgezeigt. Die Darstellung logischer Funktionen und auch negativer Zahlen kann experimentell gefunden werden. Die bisher angesprochenen Funktionsgruppen des Prozessors sind in **Bild 2** zu sehen.

## 4 Anwahl des Hauptspeichers

Die bisher als Speicher verwendeten Arbeitsregister im Mikroprozessor werden durch eine geeignete Codierung von 3 der acht Bit einer Registerinstruktion unterschieden. Zur Anwahl von außerhalb des Mikroprozessors liegenden Speicherplätzen steht dagegen ein **Adreßvolumen** von insgesamt 16 Bit zur Verfügung, die in zwei Byte der entsprechenden Instruktion nachgestellt werden.

Die drei Bytes einer mit Adresse versehenen Instruktion, z.B. „Speichere Akkumulatorinhalt auf die in folgenden zwei Bytes angegebene Adresse", werden nacheinander jeweils mit Ready-Signal vom Datenbus in den Prozessor geholt. Das erste Byte gelangt ins Befehlsregister und wird decodiert, das zweite Byte kommt anschließend als untere acht Bit der Adresse in ein insgesamt 16 Bit enthaltendes Datenadreßregister. Mit dem dritten Ready-Signal übernimmt der Mikroprozessor das dritte Byte als obere acht Bit der Adresse ebenfalls noch in sein Datenadreßregister, gibt dessen Inhalt auf einen Adreßbus (16 Bit) und den Inhalt des Akkumulators auf den Datenbus. Ein Steuersignal vom Mikroprozessor (MEMW: Memory Write) veranlaßt unmittelbar darauf den über den Adreßbus angewählten Speicherplatz, die auf dem Datenbus liegende Information zu übernehmen, d.h. einzuschreiben. Anschließend erwartet der Mikroprozessor wieder die nächste Instruktion auf seinem Datenbus.

Entsprechend läßt sich mit einer anderen Instruktion auch der Akkumulator mit dem Inhalt eines bestimmten Speicherplatzes laden. Nach dem dritten Ready-Signal veranlaßt ein vom Mikroprozessor erzeugtes Steuersignal (MEMR: Memory Read), daß der Inhalt des über den Adreßbus angewählten Speicherplatzes auf den Datenbus ausgegeben wird und von dort in den Akkumulator gelangt. Speicherplätze, die ein

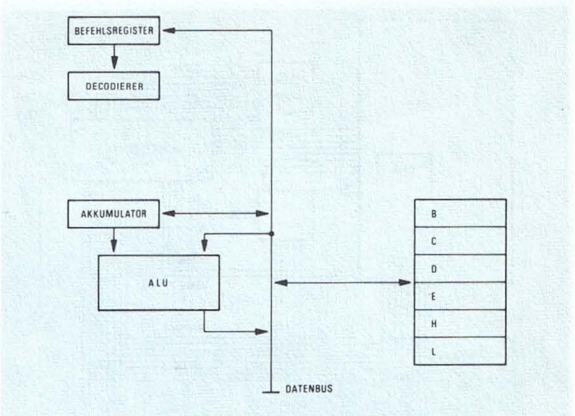



**Bild 2** Einige Funktionsgruppen im Mikroprozessor

Einschreiben und Auslesen gestatten sollen, werden mit geeigneten RAM-Bautsteinen (**RAM: Random Access Memory**) bestückt. Die bisher angesprochenen Baugruppen des Computers sind in **Bild 3** zu sehen.

Mit 16 Leitungen im Adreßbus kann der gesamte Speicher mit insgesamt $2^{16} = 65536$ verschiedenen Speicherplätzen ausgestattet sein. Die möglichen Codierungen der Adressen werden gern in einzelne Bereiche zu je zehn oder auch je acht Bit aufgeteilt. Im letzten Fall ergeben die im dritten Teil der Instruktion stehenden oberen acht Bit der Adresse jeweils gleich die Codierung des Adressenbereiches.

## 5 Ablauf eines im Hauptspeicher stehenden Programms

Im ersten, bisher angedeuteten Teil des Praktikums werden die einzelnen Instruktionen bzw. Instruktionsteile eines kurzen Programms vom Lernenden von Hand in zeitlicher Reihenfolge auf den Datenbus gebracht und unmittelbar vom Mikroprozessor verarbeitet. Im Gegensatz dazu kann das komplette Programm auch im Hauptspeicher stehen und dann daraus vom Prozessor schrittweise wieder über den Datenbus gelesen und verarbeitet werden. Die richtige Anwahl der das Programm enthaltenden Speicherplätze über den Adreßbus erledigt ein im Mikropro-

**ALU:** Arithmetic Logic Unit. Arithmetisch-logische Einheit, also das Rechenwerk.

**Adreßvolumen:** Bereich der möglichen Speicheradressen z.B. $2^{16} = 65\,536 = 64$ K, wenn der Adreßbus 16 Bit breit ist (Adreßraum).

**RAM:** Random Access Memory, Schreib-Lesespeicher; zumeist zur Abspeicherung von Daten und Ergebnissen benutzt.

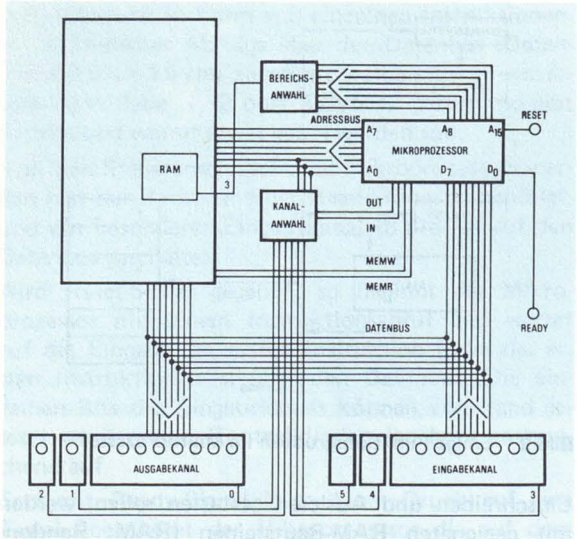



**Bild 3** Einige Baugruppen des Mikrocomputers

zessor enthaltener **Befehlszähler**, der nach jeder gelesenen Instruktion bzw. nach jedem gelesenen Instruktionsteil (zu 8 Bit) eine Adresse weiter zählt.

Der Inhalt des Befehlszählers wird mit Reset-Signal gelöscht, d.h. der Mikroprozessor wählt beim ersten Instruktionsabruf den Speicherplatz auf Adresse 0 an, anschließend 1, 2 usw. Beim LH-Mikrocomputer liegen auf den ersten Adressen die eingangs erwähnten Programmschalter (zwölf Schiebeschalter zu je 8 Bit), die bisher alle auf Null stehen. Somit erkennt der Prozessor in diesen Bytes jeweils die Instruktion „Keine Ausführung" (no operation), zählt den Befehlszähler rasch (innerhalb etwa 100 µs) über diesen Bereich weg und erlaubt dann die Instruktionseingabe von Hand auf Datenbus, siehe oben.

In der Stellung dieser Programmschalter lassen sich aber auch die Instruktionen eines kurzen Programms fixieren, z.B. die Addition zweier Zahlen und Ausgabe des Ergebnisses. Wird nun Reset-Signal gegeben, wird dieses Programm automatisch und im allgemeinen ohne Unterbrechung abgearbeitet. Auch Programmschleifen mit bedingten oder unbedingten Sprunginstruktionen sind möglich.

Längere Programme sind hier natürlich nicht unterzubringen. Die einzelnen Bytes werden dann über einen Eingabekanal (8 Bit, Tastatur oder Bandanschluß) in den Akkumulator des Mikroprozessors gebracht und von dort in die einzelnen Plätze des Hauptspeichers eingeschrieben (RAM-Bereiche, siehe oben). Dazu werden für jedes abzuspeichernde Byte etliche Hilfsinstruktionen abgearbeitet. Beim Versuch, diese wiederum alle von Hand in zeitlicher Reihenfolge auf den Datenbus zu geben, wird man schnell auf die Idee kommen, diese sich immer wiederholenden Hilfsinstruktionen in Form einer einfachen Programmschleife in den Programmschaltern zu fixieren. So hat der Lernende selbst die Notwendigkeit eines Hilfsprogramms (zur Programmeingabe) erkannt, ein solches einfaches Monitorprogramm entworfen, in einem (von Hand programmierbaren) Festwertspeicher fixiert und anschließend natürlich auch erprobt.

Mit einfachen Programmen kann man spätestens jetzt die letzten wesentlichen Funktionsgruppen des Mikroprozessors, das **Flagregister** und den **Stapelzeiger**, in ihrem Verhalten und ihrer Anwendung untersuchen. In **Bild 4** sind zu allen angesprochenen Funktionsgruppen des Mikroprozessors noch die **Taktverarbeitung** und ein **Mikroprogrammwerk** eingezeichnet, die den prozessorinternen Ablauf steuern.

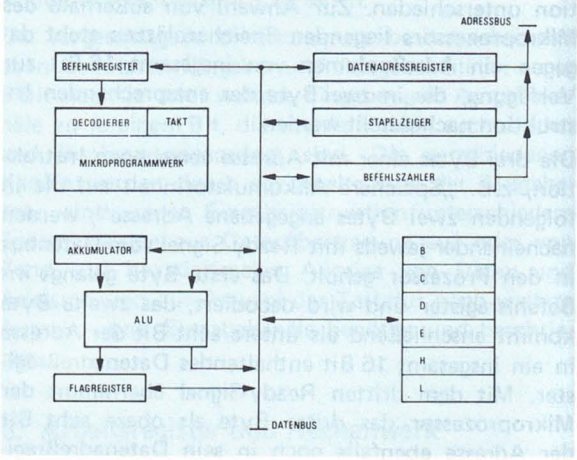



**Bild 4** Die Funktionsgruppen des Mikroprozessors

**Befehlszähler:** Enthält die Adresse des Befehls, der als nächster auszuführen ist. Auch: Programmzähler (PC: Program Counter).

**Flagregister:** Enthält eine Reihe von Bits, die Auskunft über wichtige logische Zustände in der Zentraleinheit geben (Statusbit, Statusregister).

**Stapelzeiger:** Spezielles Register, das angibt, welche Speicherstelle im Stapelbereich (Stack) als nächste belegt werden kann.

**Taktverarbeitung:** Grundprinzip moderner Datenverarbeitung, nämlich das Zusammenwirken aller Funktionseinheiten einer DV-Anlage mit einem festen Systemtakt (Taktfrequenz, z.B. 2 MHz).

**Mikroprogrammwerk:** Funktionseinheit mit Festwertspeicher (ROM), in dem Programme stehen, die den Befehlsablauf steuern.

## 6 Monitorprogramm

Der Umgang mit dem primitiven in den Programmschaltern abgespeicherten Hilfsprogramm zur Programmeingabe über einen 8-Bit-Eingabekanal läßt schnell den Wunsch nach komfortableren Monitorprogrammen aufkommen, die die Prozedur erleichtern und insbesondere gemachte Fehler leichter erkennen und korrigieren lassen. Der Mikrocomputer enthält einen ROM-Baustein (**ROM: Read Only Memory**), der auf z.B. 2048 Plätzen eine Reihe festabgespeicherte Monitorprogramme trägt, die jetzt mit einem geeigneten Sprungbefehl in den ersten drei Programmschaltern angewählt werden können.

Zunächst taucht das selbst erstellte primitive Hilfsprogramm wieder auf, um den Einstieg in die *"Black-Box"* des ROM-Bausteins zu erleichtern. Es folgen zwei weitere noch einfache Zwischenschritte (Programmeingabe mit binärer und hexadezimaler Adressenanzeige), bevor Programmeingabe, Lesen und Korrektur über eine hexadezimale Tastatur und hexadezimale Anzeige der Adressen und Speicherinhalte möglich wird. Weitere Programmteile zur Verschiebung von Datenbereichen, aufwärts und abwärts, zum Programmablauf unter Monitorkontrolle, Einzelschritt, zum Abspeichern auf und Laden von Band, auch mit automatischer Aussteuerung, werden gegeben.

## 7 Weitere Anwendungsschwerpunkte

Ein wichtiger Schwerpunkt der Grundausbildung ist sicher die schrittweise Erfassung der wesentlichsten Strukturen und Funktionen von Mikroprozessor und Mikrocomputer. An diesen Gang sei noch einmal erinnert. **Bild 5** zeigt dazu ein Foto des kompletten Mikrocomputers, auf dem vorne die mit Buchsen und Leuchtdioden ausgeführten Ein- und Ausgabekanäle, rechts die Steuerleitungen, hinten im dunklen Feld die erst zuletzt verwendeten Einheiten, wie Tastatur, hexadezimale Ziffernanzeige, Kassettenanschluß und ein Pegelumschalter (siehe unten) zu sehen sind.

Links befinden sich in zwei Kolonnen die 12 Programmschalter als Teil des Hauptspeichers, dessen Struktur zusammen mit Mikroprozessor, Adreß- und Datenbus in der Mitte als symbolischer Aufdruck erscheint.

Ausgestattet mit einem kompletten Mikrocomputer und der experimentell erworbenen Kenntnis seiner Struktur und Arbeitsweise können nun zur Ergänzung der Ausbildung oder des Selbststudiums noch weitere Schwerpunkte gesetzt werden. Für zukünftige Soft-

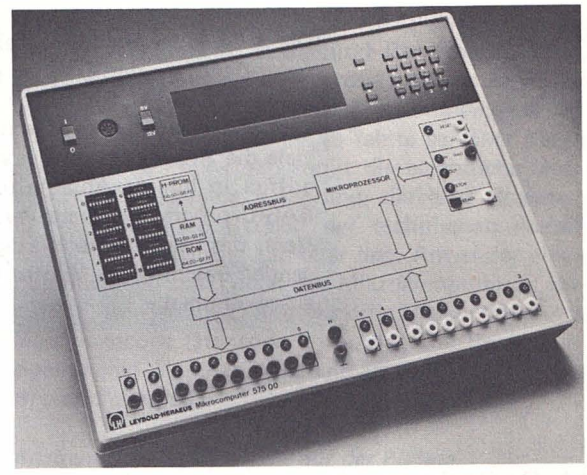



**Bild 5** LH-Mikrocomputer Kat.-Nr. 575 00 (Firma Leybold-Heraeus, Köln)

warespezialisten stehen jetzt z.B. alle Möglichkeiten offen, Programme zu erstellen, zu testen und Programmstrukturen zu erfassen.

Anwendungsfetischisten werden zahlreiche Steuerungsaufgaben suchen und deren Lösung finden. Mit den in 4-mm-Buchsen herausgeführten, kurzschlußfesten und überspannungssicheren Ein- und Ausgabekanälen stehen geeignete Steuerleitungen zur Verfügung, um externe Informationen oder Rückmeldungen aufzunehmen und einfache Maschinen oder Anlagen zu steuern. Ein Pegelumschalter für die Ausgabekanäle erlaubt den wahlweisen Anschluß von 12-V-Logik oder 5-V-Logik und anderen elektronischen Bausteinen. Damit stehen viele, experimentell leicht zu verwirklichende Möglichkeiten offen. Das Programm für eine verkehrsabhängig gesteuerte Ampelanlage ist sogar bereits im ROM-Baustein des Mikrocomputers fest abgespeichert.

Nicht zuletzt bleibt dem künftigen Hardwarespezialisten die Möglichtkeit, den Mikrocomputer nach eigenen Vorstellungen zu erweitern. Alle Leitungen von Daten-, Adreß- und Steuerbus, die Stromversorgung, Takt und einige bereits decodierte Adressen für weitere Ein-/Ausgabekanäle stehen an einem Anschluß für Vielfachstecker an der Leiterplatte des Computers zur Verfügung.

**ROM:** Read Only Memory, Festwertspeicher. Kann nur gelesen werden; meist zur Abspeicherung von fertigen (ausgetesteten) Programmen benutzt.

Der KIM (oder ein ähnlich einfacher $\mu$C) ist oftmals für inzwischen versierte Computer-Benutzer der Einstieg gewesen. Bei intensiver Nutzung kam jedoch bald unweigerlich der Wunsch nach Erweiterungsmöglichkeiten und – vor allem – nach dem „Absprung" vom Maschinencode. Der PET (oder vergleichbares) war dann für viele die Alternative. Dabei verlor man aber weitgehend alle Möglichkeiten von Eingriffen in interne Abläufe und des Studiums dieser Abläufe. Denn dafür sind in BASIC programmierbare *Personal Computer* auch gar nicht ausgelegt. Ein *Lernsystem,* das die Assembler-Eingabe mit symbolischen Adressen erlaubt, wird im folgenden beschrieben. Man ist damit einen wesentlichen Schritt über den „KIM" hinausgekommen.

M. Gottlob

# Das TI-Lern- und -Lehrsystem TM 990/189

## 1 Erfahrungen mit dem TI-Lern- und -Lehrsystem

Der große Erfolg auf dem Gebiet der Taschenrechner ermutigte TI, auch entsprechende Anstrengungen für die Entwicklung eines Lern- und Lehrsystems für das bessere Kennenlernen und Verständnis von Mikroprozessoren und Mikrocomputern zu machen. Selbstverständlich liefen umfangreiche Tests an Universitäten, Fachhochschulen, Fachoberschulen und beruflichen Ausbildungsstätten, die das TI-Lern- und -Lehrsystem nicht nur auf technische Qualität sondern insbesondere auch auf didaktische Eignung überprüften.

Durchweg wurde das TI-Lern- und -Lehrsystem mit „für die Ausbildung bestens" geeignet beurteilt. Darüber hinaus wird es häufig in der Industrie für kleinere Entwicklungen benutzt. Die Erfahrungen hieraus zeigen, daß ein großer Teil von Echtzeit-Problemen in der Meß- und Prozeßtechnik mit diesem System realisiert werden können.

## 2 Einsatzbereiche für das TI-Lern- und -Lehrsystem

- *Universitäten und Fachhochschulen*
  Hier insbesondere die Fachbereiche Mathematik, Physik, Informatik, Elektrotechnik, Datenverarbeitung und Nachrichtentechnik, Steuerung- und Regelungstechnik, allgemeine Prozeßtechnik, Meßtechnik.
- *Wissenschaftliche Institutionen* (z.B.)
  Max Planck Gesellschaft (MPG)
  Deutsche Forschungs- und Versuchsanstalt für Luft und Raumfahrt (DFVLR)
- *Berufliche Weiterbildung – allgemeine Erwachsenenbildung*
  Ausbildungsstätten der Industrie- und Handelskammern
  Ausbildungsstätten der Handwerkskammern (Berufsbildungszentren)
  Ausbildungsstätten der Innungen
  Ausbildungsstätten der Volkshochschulen
  Ausbildungsstätten anderer privater Träger
- *Industrieller Einsatz*
  Interne Schulungsstätten der Industrie
  Kleinere Entwicklungsfirmen auf den Gebieten Meßtechnik, Prozeßtechnik und speziellen Bereichen des Maschinenbaus
  Entwicklungslabors von Großfirmen
- *Schulbereich*
  Gymnasien – obere Klassen
  Fachoberschulen
  Berufsaufbauschulen

Berufsschulen
Technikerschulen
Bundeswehrfachschulen
Fachakademien

- *Hobbymarkt – Studenten*

engagierte Hobbyisten
Programmierclubs usw.
Fachstudenten für privaten Gebrauch

## 3 Technischer Aufbau

- 16 bit Mikroprozessor TMS9980A
- 1024 byte RAM, erweiterbar auf der Platine auf 2048 byte
- 4096 byte ROM oder EPROM (TMS2532), erweiterbar auf der Platine auf 6144 byte
- 16 programmierbare Ein/Ausgabe-Ports
- 4 ansteuerbare Lampen auf der Platine
- 1 ansteuerbarer Lautsprecher auf der Platine
- Eingabe in Assembler mit **symbolischen Adressen**
- Speichermanipulation in Maschinencode
- Alphanumerische Tastatur
- Anschluß für handelsüblichen Kassettenrecorder
- Anschluß für **RS232C**-Terminals und Teletype (**20 mA Linienstrom**)
- Spannungsversorgung + 5V, ± 12V
- 10stellige Anzeige
- erweiterbar durch herausgeführten Adreß- Daten- und Steuerbus
- **Baudrate:** 110 Baud und 300 Baud
- Unterprogrammverschachtelung möglich
- Einzelbefehlsausführung
- programmierbare Ein/Ausgabe erweiterbar
- Übertragungsart bei Kassettenbetrieb 0 ≙ 1200 Hz; 1 ≙ 2400 Hz
- 17 Befehle für das Betriebssystem (Monitorbefehle)

## 4 Didaktische Besonderheiten

*Die Tastatur*

Um dem Anwender am Anfang die „Bedienungsscheu" zu nehmen, wurde bewußt das Tastenfeld eines Taschenrechners verwendet. Dieses Tastenfeld ist jedem vertraut, es bedarf keiner Umgewöhnungszeit, die Einarbeitung erfolgt somit sicherer und schneller.

*Assembler-Eingabe und Maschinencode-Anzeige*

Ein wesentliches Merkmal der TI-Lern- und -Lehrplatine ist die Programmeingabe in Assembler mit symbolischen Adressen. Dadurch erhält der Anwender die notwendige Praxisnähe und Berufsbezogenheit. Er programmiert wie ein „Profi", so wie es in der Industrie üblich ist. Wäre die Programmeingabe nur im Maschinencode möglich, so fehlte jeglicher Bezug zur täglichen Praxis und somit die eigentliche Motivation. Dies ist eine Erfahrung, die ganz allgemein für den gesamten Bereich der technischen Ausbildung gemacht wurde. Um aber dem Anwender einen „Einblick" in das Leben der Mikrocomputer zu vermitteln, werden auf Tastendruck die Speicherinhalte in Maschinencode (**Hexcode**) angezeigt. Der Bezug zum eigentlichen Vorgang innerhalb des Mikrocomputers ist somit gegeben.

*Ansteuerbare Signalorgane auf der Platine*

Vier ansteuerbare Lämpchen und ein ansteuerbarer Lautsprecher befinden sich auf der Platine. Der Anwender kann – ohne großen zusätzlichen Hardware-Bedarf – den pädagogisch notwendigen „Aha-Effekt" erleben und mit Freude weiter arbeiten.

*Bedienungsfreundlichkeit* – Monitor –

Mit Hilfe von einfachen Kommandos kann eine der 17 Betriebsarten gewählt werden. Über die Anzeige und mit Hilfe von Signaltönen und Lautsprecher meldet der Mikroprozessor, daß er die entsprechenden Kommandos akzeptierte, sich in einer bestimmten Betriebsart befindet oder die Kommandos nicht akzeptiert, da fehlerhaft eingegeben wurde.

So meldet sich der Mikroprozessor z.B. nach dem Einschalten und einem internen **Selbstcheck** mit „CPU ready". Nach Betätigung der RET-Taste meldet

**Symbolische Adresse:** Vom Programmierer frei gewählte und leicht merkbare Benennung für einen bestimmten Speicherplatz. Sie muß vor dem Programmstart vom Übersetzungsprogramm in eine *absolute Adresse* umgesetzt werden.

**RS 232C:** Nummer des von der amerikanischen *Electronic Industries Association* (EIA) herausgegebenen Normblattes zur V.24-Schnittstelle.

**20 mA Linienstrom:** Strom an der sehr weit verbreiteten seriellen Strom-Schnittstelle (TTY-Schnittstelle).

**Baudrate:** Gibt an, wieviele Stromschritte pro Sekunde auf einer Leitung übertragen werden (Schrittgeschwindigkeit oder Telegrafiergeschwindigkeit). Bei serieller Übertragung gibt die Baudrate (Bd) auch die Anzahl der übertragbaren bit/s an.

**Hexcode:** Hexadezimalcode, also Darstellung im Sechzehnersystem (4 bit ergeben eine Hexadezimalstelle).

**Selbstcheck:** Automatische Selbstüberprüfung.

sich der Mikroprozessor mit einem Fragezeichen – d.h. „Zur Aufnahme von Kommandos bereit". Es ist somit eine echte Kommunikation zwischen Mikroprozessor und Anwender möglich.

## 5 Das Lehrmaterial

*Anwenderhandbuch*

Es zeigt sehr detailliert und ausführlich alle durchführbaren Funktionen, Schaltpläne und Betriebsarten auf. Die rund 150 Seiten sind übersichtlich gegliedert und mit zahlreichen Skizzen versehen. Der Anwender findet sehr schnell die notwendigen Beschreibungen, Blockschaltbilder und Kommandos, um die gewünschte Betriebsart zu verwirklichen.

*Lehr- und Übungsbuch*

Das rund 500 Seiten starke Lehr- und Übungsbuch wurde von Texas Instruments (in enger Zusammenarbeit mit anderen didaktisch erfahrenen Unternehmen) für das TI-Lern- und -Lehrsystem erarbeitet. Von einfachen Befehlserklärungen über kleine praktische Programmierübungen bis zur komplexen Unterprogrammverschachtelung und Interruptverarbeitung wird der Anwender zur sicheren Programmierung und Beherrschung des Systems geführt. Die Betonung liegt auf Übungen – nur so läßt sich genügende Sicherheit vermitteln. Der Anwender ist anschließend in der Lage, größere Programme selbständig zu schreiben und mit dem System in Verbindung mit externen Geräten zu arbeiten.

*Befehlskarte*

In Brusttaschenformat – jederzeit verfügbar, um schnell das „Wesentliche" nachlesen zu können, wenn „vor Ort" gearbeitet wird.

## 6 Erweiterungsmöglichkeiten

*Interface-Schaltung für Kassettenrecorder auf Platine*

Der Anschluß an einen Kassettenrecorder ist bereits standardmäßig vorgesehen. Es sind nur die notwendigen Brücken laut Belegungsplan zu löten, und nahezu jeder handelsübliche Kassettenrecorder kann als „externer Speicher" verwendet werden.

*Vorbereiteter Anschluß für RS232C-Terminals oder Teletype*

Mit nur wenigen zusätzlichen Bausteinen ist die Platine für die Kommunikation mit externen Terminals betriebsbereit zu machen.

*Daten- und Adreßbus sowie*
**CRU**- *und Steuerleitungen herausgeführt*

Dies ist ein wesentlicher Vorteil, da mit Hilfe von Steckern sehr einfach und schnell das System erweitert werden kann. Die Erweiterung der Speicherkapazität wie auch die der programmierbaren Ein-Ausgabe sind somit problemlos zu realisieren. Darüberhinaus bieten die herausgeführten Busanschlüsse gute Möglichkeiten für schulische Demonstrationszwecke. Mit Hilfe von entsprechenden Logikanalysatoren bzw. Oszilloskopen kann dem Anwender das jeweilige **Impulsmuster** auf dem Bildschirm gezeigt werden, was das System sehr transparent macht.

## 7 Bausteine zur Erweiterung der Funktionen und Speicher

Bauteile-Liste für Erweiterungen der TM990/189 Lern-, Lehr- und Entwicklungsplatine:

| | |
|---|---|
| ON BOARD EPROM ERWEITERUNG | U32 = TMS 2708JL ODER 2716JL |
| ON BOARD RAM ERWEITERUNG | U21 = TMS 4045-45NL |
| " " " " | U23 = TMS 4045-45NL |
| OFF BOARD MEMORY ERWEITERUNG | U18 = 74LS244 |
| " " " " | U26 = 74LS244 |
| " " " " | U27 = 74LS245 |
| " " " " | U30 = 74LS126 |
| " " " " | R34 = 10K, 5%, 0,25W |
| TERMINAL OPTION (TTY + RS 232C) | U31 = TMS 9902NL |
| " " (TTY + RS 232C) | U37 = 75 188 |
| " " (TTY) | Q12 = 2N 2905 |
| " " (TTY) | CR9 = 1N 914B |
| " " (TTY) | R32 = 33K, 10%, 0,25W |
| " " (TTY) | R33 = 3,3K, 10%, 0,25W |
| " " (RS 232C) | R35 = 3,3K, 10%, 0,25W |
| " " (RS 232C) | R36 = 3,3K, 10%, 0,25W |
| " " (TTY) | R37 = 330, 10%, 0,5W |
| " " (TTY) | R38 = 560, 10%, 0,5W |
| " " (TTY) | R39 = 2,7K, 10%, 0,25W |
| " " (TTY + RS 232C) | P3 = AMP 205858-2 ODER CANNON DBU-255-AA |
| ON BOARD KASSETTENANSCHLUSS | CR12 = 1N914B |
| " " " " | K1 = COSAR 232951-1 |
| SYSTEM FREQUENZ MODIFICATION | R49 = 10, 10%, 0,25W |
| OFF BOARD DATEN, ADRESSEN CRU | P4 = 40POLIGER STECKER |
| E/A STECKER | P5 = 40POLIGER STECKER |
| KASSETTENANSCHLUSS + E/A OPTION (TTY RS 232C) | P2 = 6POLIGER STECKER |

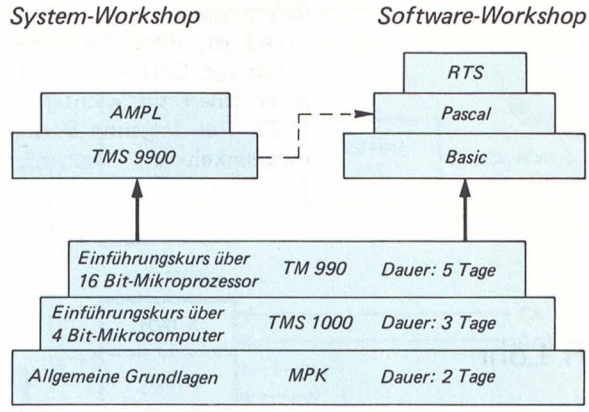
**CRU:** Programmierbare Ein/Ausgabe (Communication Register Unit)

**Impulsmuster:** Darstellung von Codeworten (Folgen von Null- und Einsbit) in Form von z.B. 5 Volt-Impulsen für Einsbit und 0 Volt für Nullbit.

## 8 Sinnvolle μP-Ausbildung – Das integrierte TI-Konzept

Es wäre falsch und nur bruchstückhaft, wenn die μP-Schulung allein auf einem System basierte. Das TI-Lern- und -Lehrsystem steht nicht allein und isoliert, sondern ist sinnvoll eingebettet in das TI-Ausbildungskonzept. Hierzu gehören vor allem die μP-Workshops von TI. Die eine Art, sich μP-Kenntnisse anzueignen, ersetzt nicht unbedingt die andere, sondern beide Methoden ergänzen sich.

*Das Workshop-Konzept von TI:*





# elrad-Triton-Computer

**Hier haben Sie endlich, wonach Sie schon lange gesucht haben: Ihren Personal Computer,**
- mit dem Sie gleichermaßen bequem in der Maschinensprache als auch in BASIC programmieren können,
- mit professioneller ASCII-Tastatur,
- mit 64 graphischen Symbolen,
- mit eingebautem UHF-Modulator,
- mit Interface für einen Kassetten-Rekorder,
- mit Gehäuse,
- mit Erweiterungsmöglichkeiten.

**Die Hardware**
- Ein-Platinen-System mit der MPU 8080 A
- Vollständige Stromversorgung
- Bis zu 3 KByte RAM auf der Platine. In der Grundausführung ist 1 K RAM enthalten.
- Interface für Kassetten-Rekorder (300 Baud), mit Motorsteuerung. Speichern und Laden von Maschinenprogrammen und BASIC-Programmen.
- UHF-Ausgang für Fernsehgerät (als Video-Monitor)
- 1 KByte RAM für alphanumerische Anzeige (memory mapping), 16 Zeilen à 64 Zeichen, ASCII-Zeichensatz und 64 graph. Symbole
- Volle ASCII-QWERTY-Tastatur (56 Tasten)
- 8 Unterbrechungsebenen
- Erweiterungsfähiger Bus (42 Anschlüsse)
- Speicher erweiterbar bis 64 KByte RAM
- 256 E/A-Ports
- Gehäuse mit Platz für Erweiterungen

**Die Software**

**Entweder . . .**

Leistungsfähiger 1-KByte-Monitor V4.1:
- Programmierung im Maschinencode
- Lesen/Schreiben von/auf Magnetband-Kassetten
- Bildschirm löschen/Cursor Reset
- Reset ohne Löschen des Speichers

2-KByte BASIC V4.1 mit folgenden Anweisungen, Funktionen und Kommandos:

| | | | | |
|---|---|---|---|---|
| ABS | IF | NEW | RETURN | STEP |
| FOR | INPUT | NEXT | RND | STOP |
| GOSUB | LET | PRINT | RUN | TO |
| GOTO | LIST | REM | SIZE | VDU |

Arithmetische und logische Operationen (ganzzahlig): Addition, Subtraktion, Multiplikation, Division, größer als, kleiner als, gleich, ungleich, größer oder gleich, kleiner oder gleich

**Oder . . .**
(für diejenigen, die gleich 'voll einsteigen' wollen)

Gegen Aufpreis liefern wir den noch leistungsfähigeren Monitor V5.1 (mit Register-Dump, Breakpoint, Continue, formatiertem Hex-Dump usw.), und eine erweiterte Version von BASIC (V5.1) mit EDIT, CALL (von Maschinenprogrammen) PEEK, POKE, READ (von beliebigen Ports), WRITE (auf beliebige PORTS), und der Möglichkeit, eigene Kommandos zu definieren.

Jeder TRITON mit V4.1-Monitor und -BASIC kann leicht nachträglich auf die V5.1-Version umgerüstet werden.

**Das Handbuch (in deutscher Sprache)**

ist verständlich geschrieben und macht Sie zum Hardwarespezialisten **und** zum Programmierer. Es hat 90 Seiten. Hier werden Sie vollständig mit allen technischen Details Ihres TRITON vertraut gemacht. Hier wird das Arbeiten mit dem Monitor ausführlich beschrieben. Und hier lesen Sie eine Einführung in das Programmieren mit BASIC (28 Programmbeispiele).

Natürlich können Sie das Handbuch auch allein erhalten. Es kostet DM 35,–.

**Zum Schluß das Beste**

Sie können alles einzeln kaufen: Platine, Gehäuse, Tastatur, Handbuch, Trafo, alle ICs, alle ROMs, alle RAMs, alle Widerstände, usw.

Welcher Personal Computer hat Ihnen das zu bieten?

**Wenn Sie noch tiefer einsteigen wollen,** dann bieten wir Ihnen an:

**Software**

Listing des Monitors und des BASIC-Interpreters V4.1
Listing des Monitors und des BASIC-Interpreters V5.1

Standard-8080-Assembler für den TRITON, auf Magnetbandkassette. Dieser Assembler setzt 3 KByte RAM im TRITON und 8 KByte RAM auf der Erweiterungskarte (s. unten) voraus.

Mit Handbuch und Auflistung des Programms.

TRITON-Standard-8080-Editor, auf Magnetbandkassette (mindestens 3 KByte RAM erforderlich). Mit Beschreibung.

**Hardware-Erweiterungen**

Das 'Motherboard' bietet Platz für 8 zusätzliche Steckplatinen im Europakarten-Format. So sind z. B. eine 8K-RAM-Karte oder eine 8-K-EPROM-Karte für eigene Programme erhältlich.

Das Netzteil und die Buffer für die zusätzlichen Platinen sind im Motherboard schon enthalten.

**Bitte beziehen Sie sich auf diese Anzeige und fordern Sie die Päckchen-Liste und Preisliste an.**

**elektronik-hobby-versand gmbh**
**An der Breiten Wiese 9**
**3000 Hannover 61**
**Tel.: (0511) 57 88 84**

elektronik hobby-versand gmbh
ehv

Für die Verbindung zwischen Mikroprozessor (CPU) und Außenwelt (Peripherie) liefern die einzelnen Hersteller eine Reihe verschiedener Interfacebausteine. Der Trend ist, diese Bausteine immer „intelligenter" zu machen, um einerseits dadurch die CPU zu entlasten, andererseits weitergehende Nutzungsmöglichkeiten zu eröffnen. Ein wichtiges Beispiel für diesen Trend ist der neue Interface-Adapter 6522. Der folgende Beitrag beschreibt ein paar äußerst nützliche Anwendungsmöglichkeiten.

R. Löhr

# Impulserzeugung mit dem R 6522 VIA

## (Versatile Interface Adapter)

Mit dem 6522 VIA ist dem Benutzer der 65xx-Systeme von Rockwell (AIM 65), Commodore (KIM-1, PET 2001) und Synertek (SYM-1) und anderer Mikroprozessoren dieser und verwandter Familien ein sehr leistungsfähiger Interfacebaustein in die Hand gegeben, der eine „intelligente" Eigenverwaltung von Dienstleistungen vornehmen kann.

Aus Platzgründen wollen wir hier keine Wiederholung aus dem Datenblatt bringen, das ja bei den Herstellern und ihren **Distributoren** bezogen werden kann, sondern einige der vielen möglichen Spezialfunktionen herausstellen. Das Blockschaltbild **Bild 1** zeigt zunächst Ähnlichkeiten zum bekannten **PIA** 6520 oder 6820 auf: Die Verbindung zur Außenwelt an den **Ports** A und B mit je 8 **Pins** wird durch Einschreiben zunächst in die *Datenrichtungsregister* DDRA und DDRB bestimmt. Das hier hinterlegte Bitmuster legt fest, welche der Pins in den *Ausgangsregistern* ORA und ORB als Ein- bzw. als Ausgänge dienen. Den gewünschten Signalpegel *High* oder *Low* für die Ausgangspins schreibt man hernach in die Register ORA und ORB.

Diese beiden Register sind gegenüber denen des PIA in Ihrer Funktion schon erweitert. Unter hoher Last mag an einem mit '1' beschriebenen Ausgangspin eine Spannung herrschen, die dem Zustand '0' entspricht. Am Port B kann man gleichwohl das ursprünglich geschriebene Signal auslesen. Und an Port A kann das dort anstehende Bitmuster in der Form zwischengespeichert werden (**Latching**), wie es vor einem aktiven Signalübergang an der *Handshake-Leitung* CA1 bestand – eine elegante Möglichkeit, die Quelle der Veränderung durch logische Befehle ausfindig zu machen.

Damit sind weitere Möglichkeiten des Bausteins angesprochen: Zu den genannten Ports treten die 4 mit CA1, CA2, CB1 und CB2 bezeichneten Steuerleitungen, deren Funktion durch initialisierendes Einschreiben in das *Peripherial Control Register,* PCR, des VIA festgelegt wird.

Mit diesen vier Steuerleitungen kann man 16 verschiedene Reaktions-Kombinationen verwirklichen, insbesondere Antworten auf aufsteigende oder abfallende Impulsflanken und bequemes Handshake mit Peripheriegeräten.

**Distributor:** (Verteiler). Großhändler oder Lieferant für Bausteine, Geräte etc. meist verschiedener Hersteller.

**PIA:** *Peripheral Interface Adapter.* Interfacebaustein mit z.B. 16 Ein- und Ausgängen, evtl. Handshake-Leitungen und oft noch ROM- und RAM-Speichern.

**Port:** Ein-Ausgang, hier mit 8 Leitungen, für 1 byte also.

**Pin:** Anschlußstift

**Latch:** Auffang- oder Haltespeicher für z.B. 1 bit oder 1 byte.

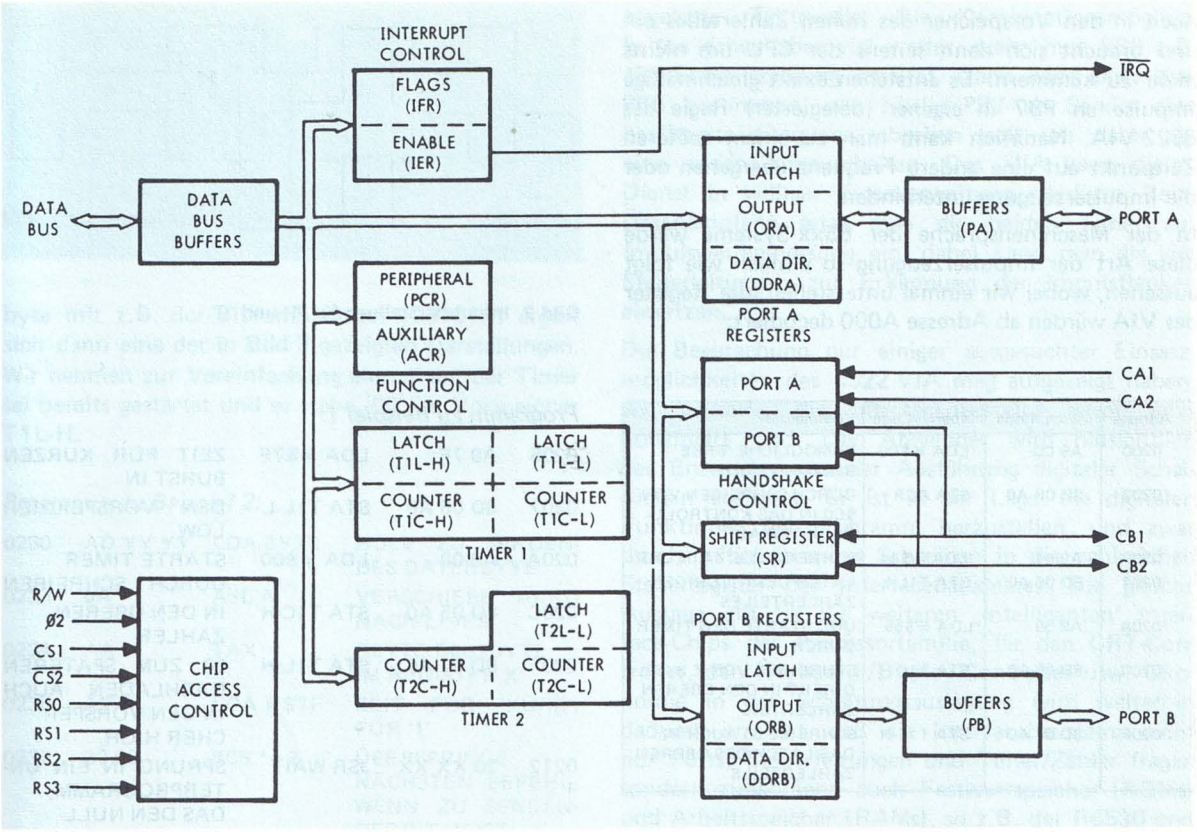



**Bild 1** Block-Diagramm des R6522 VIA (Erläuterungen im Text)

Unter der Themenstellung sind die für Spezialfunktionen ausgelegten Pins PB7 für die Impulsausgabe und PB6 für die serielle Datenübertragung sowie für die Ereigniszählung interessant. Bei serieller Übertragung arbeitet PB6 mit dem auf dem Chip enthaltenen *Schieberegister* SR zusammen. PB7 und PB6 können in Verbindung mit den ebenfalls vorhandenen **Timern** T1 und T2 betrieben werden.

Beide Timer sind geschickt angelegt. Sie arbeiten in 16 Bit Breite, haben also ein ‚hohes' und ein ‚niedriges' Byte, unter dem der Zählerstand erfaßt werden kann. Der besondere Wert des Timers T1 ergibt sich aber aus dem Vorhandensein von 2 Vorspeicher-Latches. Man kann also zu irgendeiner Zwischenzeit, während der der Timer noch herunterzählt, bereits festlegen, mit welchem Wert er nach seinem Nulldurchgang automatisch nachgeladen werden soll. Diese Betriebsweise des **Free Running Mode** erlaubt es weiterhin, bei jedem Nulldurchgang des Timers eine automatische Phasenumkehr an PB7 als Ausgangspin hervorzurufen. Und damit wird schon klar, wie vielseitig die Impulserzeugung mit dem VIA sein kann:

Man lädt zunächst den Vorspeicher des niedrigen Zählerteils mit dem gewünschten Wert und startet dann den Timer durch Schreiben direkt in den hohen Zähler. Dadurch wird zugleich auch das niedrige Latch in den niedrigen Zähler übertragen, der Timer arbeitet jetzt an der 16 Bit breiten Vorgabe. Nun kann man entscheiden, ob an PB7 immer exakt gleiche Impulse weiterlaufen sollen oder ob die nächste Phase eine andere Länge haben soll. Im ersten Fall schreibt man den benötigten Wert also auch

**Timer:** Zeitgeber. Funktionseinheit, die nach Anstoß (*Initialisierung*) mit einem vorgebbaren Takt abläuft und Steuerimpulse abgibt.

**Free Running Mode:** Freilauf. Z.B. selbsttätiger Ablauf eines einmal angestoßenen Timers.

noch in den Vorspeicher des hohen Zählerteiles ein und braucht sich dann seitens der CPU um nichts mehr zu kümmern: Es entstehen exakt gleichmäßige Impulse an PB7 in eigener (delegierter) Regie des 6522 VIA. Natürlich kann man zu einem späteren Zeitpunkt auf eine andere Frequenz übergehen oder die Impulserzeugung unterbinden.

In der Maschinensprache der 65xx-Systeme würde diese Art der Impulserzeugung so einfach wie folgt aussehen, wobei wir einmal unterstellen, die Register des VIA würden ab Adresse A000 decodiert:

| Adresse | Masch.-Code | Mnemo-Code | Kommentar |
|---|---|---|---|
| 0200 | A9 C0 | LDA #$C0 | ERMÖGLICHE 'FREE RUNNING MODE' |
| 0202 | 8D 0B A0 | STA ACR | DURCH SCHREIBEN VON $C0 IN DAS KONTROLL-REGISTER ACR |
| 0205 | A9 44 | LDA #$44 | SCHREIBE Z.B. 44 IN DAS |
| 0207 | 8D 06 A0 | STA T1L-L | LATCH DES NIEDRIGEN ZÄHLERTEILES |
| 020A | A9 55 | LDA #$55 | UND STARTE DEN TIMER DURCH |
| 020C | 8D 05 A0 | STA T1C-H | SCHREIBEN VON Z.B. 55 DIREKT IN DEN OBEREN ZÄHLERTEIL |
| 020F | 8D 07 A0 | STA T1L-H | SCHREIBE 55 AUCH IN DAS LATCH DES OBEREN ZÄHLERTEILS |

Soll die zweite Impulsphase dagegen eine andere Länge haben, so wird man im Verlaufe entsprechende Werte in die Vorspeicher schreiben. Man braucht also nicht zu warten, bis der Zähler seinen Nulldurchgang hat und dabei ein Flag setzt.

Bei der Anwendung dieser elegangen Möglichkeiten brauchen wir nicht nur an frequenzgesteuerte Motoren zu denken, naheliegend ist vor allem auch die Datenübertragung und die Datenspeicherung auf magnetischen Trägern. Die Datenspeicherung auf Magnetband mag dabei frequenzcodiert sein oder irgendeine andere beliebige Darstellung in Rechteckimpulsen haben. Hierzu zwei schematische Instruktionsfolgen, die man in der Praxis als Programmschleifen anlegen wird.

*Im ersten Beispiel* beginnt ein Datenbit immer mit einer kurzen Halbwelle als **Burst**. Die zweite Halbwelle ist kurz, wenn es sich um eine binäre '1' handelt, und sie ist doppelt so lang für eine '0' (**Bild 2**). *Das zweite Beispiel* betrifft Wechselphasencodierung. Nicht mehr der **Signalpegel** ist für die Darstellung von binär '0' oder '1' erheblich, sondern allein die Zeit zwischen den Phasensprüngen. Für ein Daten-

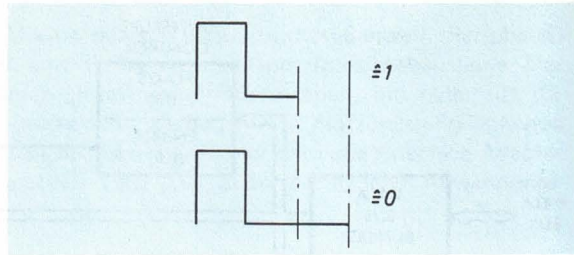



**Bild 2** Impulsdarstellung für '1' und '0'.

*Programm zu Beispiel 1:*

| | | | |
|---|---|---|---|
| 0205 | A9 7F | LDA #$7F | ZEIT FÜR KURZEN BURST IN |
| 0207 | 8D 06 A0 | STA T1L-L | DEN VORSPEICHER LOW |
| 020A | A9 00 | LDA #$00 | STARTE TIMER DURCH SCHREIBEN |
| 020C | 8D 05 A0 | STA T1C-H | IN DEN OBEREN ZÄHLER |
| 020F | 8D 07 A0 | STA T1L-H | 00 ZUM SPÄTEREN NACHLADEN AUCH IN DEN VORSPEICHER HIGH. |
| 0212 | 20 XX XX | JSR WAIT | SPRUNG IN EIN UNTERPROGRAMM, DAS DEN NULLDURCHGANG DES TIMERS FESTSTELLT |
| 0215 | AD YY YY | LDA BYTE | LADE ZU SENDENDES DATENBYTE |
| 0218 | 0A | ASL A | LINKSVERSCHIEBE UM EINE BITPOSITION INS 'CARRY' |
| 0219 | AA | TAX | RETTE RESTBYTE IN REGISTER X |
| 021A | A9 7F | LDA #$7F | TIMERZEIT FÜR 'KURZ' |
| 021C | B0 02 | BCS* + 2 | ÜBERSPRINGE NÄCHSTEN BEFEHL, WENN DAS BIT EINE '1' WAR |
| 021E | A9 FE | LDA #$FE | LADE DOPPELTEN WERT FÜR '0' |
| 0220 | 8D 06 A0 | STA T1L-L | IN DEN VORSPEICHER LOW |
| 0223 | 20 XX XX | JSR WAIT | WARTE BIS ZUM PHASENSPRUNG |

**Burst:** Impuls, Stoß

**Pegel:** Logarithmiertes Größenverhältnis, meist in *Dezibel* (dB) angegeben.

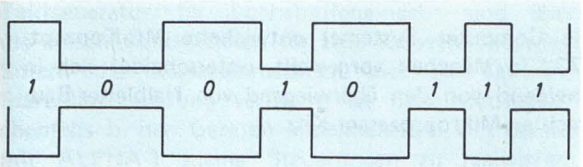



**Bild 3** Wechselphasen-Codierung

byte mit z.B. der Bitkombination 10010111 ergibt sich dann eine der in **Bild 3** gezeigten Darstellungen. Wir nehmen zur Vereinfachung einmal an, der Timer sei bereits gestartet und es stehe '00' im Vorspeicher T1L-H.

*Programm zu Beispiel 2:*

| | | | |
|---|---|---|---|
| 0230 | AD YY YY | LDA BYTE | HOLE ZU SENDENDES DATENBYTE |
| 0233 | 0A | ASL A | VERSCHIEBE AKKU NACH LINKS |
| 0234 | AA | TAX | RETTE RESTBYTE IM REGISTER X |
| 0235 | A9 7F | LDA #$7F | ZEIT FÜR 'KURZ', FÜR '1' |
| 0237 | B0 02 | BCS *+ 2 | ÜBERSPRINGE NÄCHSTEN BEFEHL, WENN ZU SENDENDES BIT '1' IST |
| 0239 | A9 FE | LDA #$FE | LADE DOPPELTEN WERT FÜR '0' |
| 023B | 8D 06 A0 | STA T1L-L | IN DEN VORSPEICHER LOW |
| 023E | 20 XX XX | JSR WAIT | ERWARTE DEN PHASENSPRUNG |

An PB7 kann man auf ähnliche Art auch einmalige Impulse (*One Shot*) definierter Länge erzeugen. Ebenso ist über PB6 eine vollautomatische serielle Datenübertragung aus dem oder in das Schieberegister möglich, wobei entweder der Timer T2 den Takt bestimmt oder wahlweise auch eine außen angelegte Taktquelle. Die Kombinationsmöglichkeiten gehen aber viel weiter, wenn man PB6 z.B. auf Ereigniszählung schaltet. Dann kann man über PB6 bestimmen, wie häufig PB7 als Sender eine bestimmte Frequenz abgeben soll um dann auf eine andere umzuschalten. Das VIA kann diesen Dienst in völliger Eigenverwaltung erledigen. Beim Datenempfang setzt man die beiden Timer zur Impulsweitenmessung ein, dabei kann man die vier Steuerleitungen zur Erkennung der Impulsflanken einsetzen.

Die Besprechung nur einiger ausgesuchter Einsatzmöglichkeiten des 6522 VIA mag aufgezeigt haben, wie vielseitig die Interfacebausteine schon jetzt konzipiert sind. Der Anwender wird hinsichtlich des Entwurfes und der Ausführung digitaler Schaltungen entlastet. Er ist in der Lage, die digitalen Funktionen per Programm herzustellen, und zwar durch initialisierendes Schreiben in die zahlreichen Steuerregister des Interfacebausteines. Die gleiche Aussage gilt für die weiteren ‚intelligenten' Interface-Chips der Prozessorfamilie, für den **CRT-Controller,** den **Keyboard-/Display-Controller** usw. Ökonomie in der Schaltungsauslegung wird weiterhin dadurch erreicht, daß die Interfacebausteine nicht nur Ports, Steuerleitungen und Timer/Zähler tragen sondern zunehmend auch Festwertspeicher (ROMs) und Arbeitsspeicher (RAMs), so z.B. der R6530 und der R6531.

**CRT:** *Cathode Ray Tube,* Elektronstrahlrohr. Gemeint ist die Bildröhre von Sichtgeräten (*Video Terminals*).

**Keyboard:** Tastatur, z.B. ASCII- oder Hexadezimal-Tastatur.

**Display:** Anzeigeeinheit, z.B. Leuchtziffern oder auch Sichtgerät.

**Controller:** Funktionseinheit, die Anpassungen durchführt und Abläufe steuert.

Das von MCS-Berlin (Micronic Computer Systeme) entwickelte MC-Konzept, erstmals auf der „SYSTEMS 77" in München vorgestellt, unterscheidet sich in Aufbau und Zielsetzung entscheidend von den überwiegend von Halbleiter-Bauelemente-Herstellern herausgebrachten Mikroprozessor-Kits.

B. Strößner

# ALPHA 1 und BETA 8 – Ein Mikrocomputersystem aus Berlin

Einziger Orientierungsmaßstab war die Entwicklung eines nach kommerziellen Maßstäben aufzubauenden Systems, dessen Grundelemente der Lern- und Entwicklungsmikrocomputer ALPHA 1 und das für den industriellen Einsatz konzipierte Modulsystem BETA 8 sind.

Beide Geräte entstanden aus der Notwendigkeit, sowohl für eigene Projekte als auch für andere Anwender ein **kompatibles System** zu schaffen, das allen Anforderungen derjenigen, die sich in die Materie einarbeiten wollen, sowie den Spezialisten genügen sollte. Durch die Unabhängigkeit von MCS, die sich ausschließlich mit der Entwicklung und Fertigung von elektronischen Geräten befaßt, war es möglich, ein zwischen technisch Möglichem und wirtschaftlich Vertretbarem optimales Verhältnis bei der Wahl der elektronischen und elektromechanischen Bauelemente zu berücksichtigen. Ein Ergebnis dieses Konzepts besteht u.a. darin, daß bis auf wenige Ausnahmen nur Bauteile eingesetzt werden, für die es mehrere Hersteller gibt.

Bei jedem **MC**-System kommt der Wahl des Mikroprozessors eine besondere Bedeutung zu. In dem BETA-8-System, obwohl nicht prozessorspezifisch ausgelegt, wird die Mikroprozessorfamilie 6500 eingesetzt. Der Einsatz des Mikroprozessors 6502, der bei der Entwicklung dieser Systeme noch relativ neu war, hat sich nicht nur wegen seiner inzwischen beachtlichen Verbreitung auch auf dem europäischen Markt (Verwendung in KIM, PET, u.a.), sondern auch wegen der ausgezeichneten Möglichkeiten im Softwarebereich als sehr sinnvoll erwiesen. Ein namhafter deutscher Konzern, der bislang einen Prozessor vom Typ 8080 bevorzugte, kam kürzlich mit einem Heimcomputer ebenfalls auf der Basis des 6502 auf den Markt.

Der ALPHA-1-Mikrocomputer, als Lern- und Entwicklungsgerät konzipiert, ist mit dem Mikroprozessor 6502 ausgerüstet. Dieser Mikrocomputer ist ein netzanschlußfertiges Gerät, welches dem Benutzer gestattet, sich in die Funktion und Programmiertechnik von mikroprozessorgesteuerten Systemen einzuarbeiten. Der Mikrocomputer ALPHA 1 ist ohne Einschränkung kompatibel zum BETA-8-System, so daß hiermit *erstmalig* ein Konzept verwirklicht wurde, das es dem Benutzer gestattet, ein Lerngerät in Verbindung mit einem industriellen System zu benutzen. Beide Geräte verfügen über eine **V24-, Strom-** und **TTL-Schnittstelle,** einen

**MC:** Mikrocomputer

**Kompatibles System:** (Austauschbarkeit, Verträglichkeit zwischen Systemen). Wenn von einem einfacheren auf ein leistungsfähigeres $\mu$C-System aufgestiegen wird, können die vorhandenen Komponenten weiterbenutzt und es muß nicht umgelernt werden.

**V24-Schnittstelle:** Serielle Übergangsstelle (*Interface*) zwischen CPU und Außenwelt (*Peripherie*) mit – bei Bedarf – diversen Steuer- und Meldeleitungen. International und nach DIN 66020 genormt.

**Strom-Schnittstelle:** (TTY-Schnittstelle oder 20-mA-Stromschleife). Nicht genormte, aber sehr weit verbreitete Serienschnittstelle ohne Steuerleitungen; zur Überbrückung großer Entfernungen geeignet.

**TTL-Schnittstelle:** Ganz allgemeine Schnittstelle, für die lediglich die TTL-Signalspannungen 0V und 5V verabredet sind.

**TTL:** *Transistor-Transistor-Logik.* Standard-Halbleitertechnologie mit nur 5V Versorgungsspannung.

Taktgenerator für Lochstreifeneingabe und über die Anschlußmöglichkeit für zwei Kassettenrecorder. Sofern die Kassettenrecorder über eine externe Start/Stopsteuerung verfügen, ist diese Betriebsart ebenfalls in den Geräten **implementiert.** Um bereits mit ALPHA 1 kleine Steuerungen zu realisieren, stehen dem Anwender zusätzlich 16 freidefinierbare Ein/Ausgabekanäle zur Verfügung. Alle Ein- bzw. Ausgabeinformationen sind an entsprechende Steckverbindungen geführt, die sich an der Frontplatte befinden.

Softwaremäßig sind sowohl ALPHA 1 als auch BETA 8 mit einem eigenen, von MCS entwickelten, *Betriebssystem* ausgestattet. Die Struktur des **2K**-Monitorprogramms MONA ist so ausgelegt, daß ein großer Teil dieser Betriebssoftware in Anwenderprogramme miteinbezogen werden kann. Alle Ein- und Ausgabeelemente werden durch das Betriebssystem gesteuert. Als Eingabeelemente stehen dem Anwender eine hexadezimale Tastatur mit 11 speziellen Funktionstasten sowie eine 8-stellige Siebensegmentanzeige für die Ausgabeinformation zur Verfügung. Durch die softwaremäßige Ansteuerung lassen sich auch Buchstaben zur Anzeige bringen. Diese Darstellungsart wird in dem ebenfalls vorhandenen *Disassembler* ausgenutzt, der eine **mnemonische Darstellung** des hexadezimalen Befehlssatzes des Prozessors erlaubt. Die internen *Register* des Mikroprozessors lassen sich per Tastendruck in der Anzeige darstellen. Weitere Betriebsarten sind **Single Step** und **Slowstep.** Hiermit wird der Anwender in die Lage versetzt, den Befehlsablauf schrittweise zu verfolgen. Auch in dieser Betriebsart ist die mnemonische Darstellung möglich. Anstelle der Tastatur und Anzeige ist eine Kommunikation in Verbindung mit einem *Datenterminal,* **Teletype** oder ähnlichem Ein- und Ausgabegerät möglich.

Zur Unterstützung aller technischen Merkmale der Hard- und Software dieser Geräte ist von den Entwicklern ein ca. 100-seitiges Handbuch zu ALPHA 1 herausgegeben worden. Außer der Beschreibung des Systems sind sämtliche Schaltunterlagen und das **Assemblerlisting** des Betriebssystems MONA vorhanden. Inzwischen wurde auch das Handbuch Band 2 mit vielen Programm- und Anwendungsbeispielen herausgegeben.

Eine gute technische Dokumentation ist die wichtigste Voraussetzung für das Arbeiten auf dem Gebiet der Mikroprozessortechnik. MCS hat es sich zur Aufgabe gemacht, gerade in dieser Hinsicht den Anwender nach besten Möglichkeiten zu unterstützen. Dies bedeutet, daß es zu jeder Baueinheit ausführliche technische Unterlagen sowohl der Hard- als auch der Software gibt. Für Lehr- und Unterrichtszwecke stehen neben ausführlichen technischen Informationsunterlagen auch **Overheadfolien** zur Verfügung. Auf besonderen Wunsch veranstaltet MCS Schulungsseminare. Im Gegensatz zu der weit verbreiteten Meinung weisen wir jedoch darauf hin, daß innerhalb eines ein- bzw. zweitägigen Seminars nur eine erste Hilfestellung im Umgang mit der Mikroprozessortechnik gegeben werden kann. Um selbst kreativ auf diesem Gebiet tätig zu sein, ist auch unter der Voraussetzung, daß optimale technische Hilfsmittel zur Verfügung stehen, ein intensives Studium dieser Materie in jedem Fall erforderlich.

Nach den kurzen Erklärungen zu dem Mikrocomputer ALPHA 1 soll nun auf die besonderen Merkmale des BETA-8-Systems eingegangen werden. Wichtigstes Merkmal ist die von MCS definierte prozessorunabhängige BUS-Struktur. Der mechanische „**Tandem**"-Grundaufbau als **19-Zoll-Tischgehäuse** *oder* **Rackeinschub** *für* **Europakarten** (100 × 160 mm) gestattet eine praxisgerechte Kartenorganisation, die die Vorzüge eines Mikrocomputersystems erst voll zur Geltung kommen lassen. Das BETA-8-System bietet dem Anwender die Möglichkeit, 16

**Implementierung:** Modisches Fachwort für „technische Ausführung" (von engl. *implement,* d.h. Werkzeug, Gerät).

**1K** = $2^{10}$ = 1024.

**Mnemonische Darstellung:** Darstellung von nur mühsam zu handhabenden Binär- oder Hexadezimalcodes mit leicht merkbaren Abkürzungen (*mnemonische Ausdrücke*).

**Single Step:** Einzelschritt, Abarbeitung nur eines einzigen Programmbefehls.

**Slowstep:** Single Step mit automatischer, aber verzögerter Weiterschaltung.

**Teletype:** (TTY). Fernschreiber.

**Listing:** Auflistung eines Programms.

**Overheadfolien:** Klarsichtfolien für Tageslichtprojektor (Overhead Projector, auch „Lichttafel").

**Tandem-Anordnung:** Anordnung mit zwei „Sitzen" (hier: Einschubfächer) hintereinander.

**19-Zoll-Gehäuse:** Standardgehäuse, 483 mm (19") breit, zum Einbau in genormte 19"-*Racks* geeignet.

**Rack:** (Gestell, Ständer). Hier: Tischgestell oder großes, oft fahrbares Gestell zur Aufnahme mehrerer 19"-Rahmen (Gehäuse).

**Europakarte:** Schaltungsplatine in den Abmessungen 100 × 160 mm, zum Einschub in 19"-Europakartenträger geeignet.

Europakarten in beliebiger Kombination einzusetzen. Der Aufbau auf dem zentralen **Motherboard** ist so organisiert, daß hiervon 5 Steckplätze für eine *rückseitige* Einschubmöglichkeit der Europakarten vorhanden sind. Diese Möglichkeit wird überwiegend dann benutzt, wenn Peripheriekarten zum Anschluß externer Steuerungen in das System eingesetzt werden und die Kabelverbindungen, wie allgemein üblich, nach hinten herausgeführt werden sollen. Im Gegensatz zu den rückseitigen Steckplätzen, deren Tiefe genau der Europakartentiefe von 160 mm entspricht, besteht für die Karten an der Frontseite eine freie Bautiefe von ca. 5 cm. Diese *freie Bautiefe* ermöglicht in Verbindung mit den Standardelementen des Systems eine individuelle Frontplattengestaltung. Dadurch ist es möglich, ein neues, vom Anwender zu definierendes Gerät zu konzipieren, ohne eine zweite mechanische und elektrische Baueinheit einsetzen zu müssen. Ein anderer Vorteil dieser 5 cm freien Bautiefe besteht darin, daß auch andere Europakarten mittels Zwischenadapter auf den Bus angepaßt werden können. Diese rein mechanischen Konstruktionsmerkmale bieten dem Anwender eine Reihe von Vorteilen, die bei anderen Modulsystemen nicht vorhanden sind.

Die Spannungsversorgungseinheit für das BETA-8-System ist ebenfalls als Kompaktbaugruppe, die rückwärts in das System eingesetzt wird, ausgebildet.

Das Netzteil mit einem Wirkungsgrad von ca. 65% liefert folgende Spannungen (Strom): +5 V (10 A); +12 V (1,5 A); −12 V (1,5 A). Alle Ausgänge sind kurzschlußfest und vor Überspannung gesichert.

Das BETA-8-System besteht aus einer Vielzahl von Komponenten, die es ermöglichen, für fast jede Problemstellung auch eine Lösung anbieten zu können.

Die *Systemkomponenten* lassen sich in vier Gruppen aufteilen:

1. Prozessorkomponente, die auch als **Single Board Computer** betrieben werden kann, mit dem Prozessor 6502, mit **I/O**, ROM-Speicher, RAM und serieller Schnittstelle, sowie eine CPU-Karte, die in Verbindung mit den anderen Komponenten betrieben wird.

2. Speicherkomponenten wie 2K-RAM, 8K-RAM, 32K-RAM und eine universelle, mit 8K bestückbare Speicherkarte für die verschiedensten RAMs, ROMs, PROMs und EPROMs.

3. Peripheriekomponenten, die wohl zahlenmäßig größte Gruppe, wie es für ein Prozessorsystem selbstverständlich ist, bilden die Verbindung des Prozessors zur „Außenwelt". **Terminal**, Drucker usw. werden ebenso zur Peripherie gezählt wie **Prozeßhardware**, intelligente Meßgeräte oder **Massenspeicher** (Floppy Disk).

Das Konzept, mit Aufsatzkarten (**Sandwichbauweise**) auf die Peripheriekarten eine individuelle Anpassung an die geforderten Anschlußdaten zu realisieren, ermöglicht es, daß nur zwei universelle digitale I/O Karten (je 32 Kanäle) benötigt werden (2 × 6520PIAs oder 2 × 6532PIAs). Eine Aufsatzkarte mit durchkontaktiertem Lochraster ermöglicht den **Prototypenaufbau.** Darüberhinaus gibt es einen Aufsatz mit 16 **optoentkoppelten** und getriebenen **Darlington Eingängen** und 16 optoentkoppelten Ausgängen sowie eine Karte mit dem Betriebssystem MONA 2, V24- oder 20-mA-Stromschnittstelle. Diese Karte hat auch den Programmsockel für das Druckerbetriebssystem. Interfacekarten für Floppy Disk, **IEC-Bus**, Drucker, serielle Schnittstellen und **Videointerface** gehören zum Programm wie *Analog-Digital-* und *Digital-Analog-Wandler.* Bedienungskonsole für hexadezi-

**Motherboard:** Mutterkarte mit Datenbus-, Steuerbus- und Versorgungsanschlüssen für mehrere Schaltungskarten (Platinen).

**I/O:** *Input/Output,* d.h. Eingabe/Ausgabe.

**Single Board Computer:** Einplatinencomputer, vollständiger Computer auf nur einer Platine (Spezialfall: EBC, d.h. *Euro Board Computer,* Europakarten-Computer).

**Terminal:** (engl. für Endstation). Hier: Datenstation, Bedienungsgerät (Kassenterminal, Sichtgerät etc.).

**Prozeßhardware:** Meßwertaufnehmer, Meßgeräte, Wandler (ADC) usw.

**Massenspeicher:** Speichermedien zur Aufnahme und langfristigen Archivierung großer Datenmengen. Bei µC meist Magnetbandkassette und Flexible Magnetplatte (Floppy Disk).

**Sandwichbauweise:** Schichtstruktur – wie beim Sandwich (belegtes Brot).

**Prototyp:** Muster, Urbild, erster Schaltungsaufbau.

**Optokopplung:** Verbindung mittels Lichtübertragung, also keine „leitende" Verbindung (galvanisch getrennt).

**Darlington-Stufe:** Kombination zweier Transistoren in der Weise, daß sie sich wie ein einzelner mit hoher Stromverstärkung verhalten.

**IEC-Bus:** International genormte Leitungsanordnung und Schnittstelle zur 8bit-parallelen Datenübertragung mit zusätzlich 8 Steuerleitungen (*Handshake-Verfahren*).

**Videointerface:** Schnittstelle zum Anschluß eines gewöhnlichen Fernsehers.

male Eingabe und Anzeige und Simulator für digitale Peripheriezustände gehören ebenso zum Lieferprogramm wie Eingangs- und Ausgangskarten für die Steuerungstechnik.

4. **Firmwarekomponenten** sind für die Entwicklung und Konstruktion von Systemen von Bedeutung. Unter diesen Begriff fallen die **EPROM**-Programmiereinheit für alle gängigen EPROM-Typen, die BUS-Kontrollkarte als Interface zum Überprüfen der BUS-Zustände, eine Laborkarte für die experimentelle Inbetriebnahme von Bausteinen und die *Prozessoremulator-Karte* zum Ersetzen des Prozessors in einem Gerät durch das BETA 8 während der Entwicklungsphase sowie ein EPROM-**Emulator,** um während der Entwicklung den Festwertspeicher durch RAM-Speicher zu ersetzen. Die Europakarten sind ausgerüstet mit der 64-poligen VG-Steckverbindung im 96-poligen Gehäuse, jeweils Reihe A und C bestückt. Alle integrierten Schaltungen sind steckbar auf **DIL-Fassungen** angeordnet. Die Vorwahl bestimmter Adreßbereiche oder andere vom Anwender auswählbare Betriebskombinationen werden ausnahmslos durch DIL-Schalter bzw. durch Steckbrücken durchgeführt. Für keine derartigen Betriebsartänderungen wird ein Lötkolben benötigt.

Neben dem Einsatz speziell des ALPHA 1 für Lehr- und Schulungszwecke in der Industrie, an Hochschulen und Instituten kommen diese Mikroprozessor-Systeme in den verschiedensten Bereichen zur Anwendung.

Die folgende Aufzählung soll dem Leser eine Vorstellung über die Vielseitigkeit der Anwendungen geben, ohne jedoch auch nur annähernd eine Vollständigkeit darzustellen:

- Steuerung von Fahrscheinverkaufsautomaten
- Klimasteuerung
- Frequenzkanalprogrammierung von Mobilfunkgeräten
- Datenaufbereitung (z.B. für den öffentlichen Nahverkehr)
- Steuerung eines audiovisuellen Lehrsystems
- Dosierungssteuerung für Betonmischwerke
- Lichtorgeln
- Fernwirkanlagen
- Projektorsteuerung
- Steuerung einer Laser-Schneidemaschine
- Waagendosiersteuerungen

**Firmware:** Von der Herstellerfirma des Computers (der Hardware also) entwickelte und in Festwertspeichern oder auf Kassetten etc. mitgelieferte Software. In diesem Fall sind auch spezielle Hardware-Komponenten gemeint.

**EPROM:** *Erasable and Electrically Programmable ROM,* also löschbarer und elektrisch programmierbarer Festwertspeicher (z.B. 2708, 2716).

**Emulator:** Programm zur Nachbildung (*Simulation*) eines Rechners auf einem anderen (verschiedenen) Rechner, wobei allerdings nur das äußere Verhalten „simuliert" wird. Im vorliegenden Fall wird die Nachbildung durch Hardware erreicht, nämlich durch die CPU des BETA-Systems.

**DIL:** *Dual-In-Line.* Prinzip der gewöhnlichen IC-Steckfassungen mit zwei nebeneinanderliegenden Steckreihen (DIL-Fassung).

Für diverse Steuerungsaufgaben und in komplett aufgebauten Mikrocomputern werden in der Regel wortorganisierte µP verwendet, Prozessoren also mit fester Wortlänge (meist 8 bit) und festem Befehlssatz. Manchmal aber reichen die möglichen Verarbeitungsgeschwindigkeiten nicht aus. Dann ist man gezwungen, auf die sehr schnellen Bitscheiben-Prozessoren (Slice Processors) zurückzugreifen. Das jedoch hat weitere Konsequenzen, z.B. Auswahl „passender" Bauteile, Kaskadierung, Mikroprogrammierung. Dieser Beitrag gibt eine detaillierte Einführung.

H.-J. Stuckenberg

# Bipolare, schnelle Mikroprozessorsysteme

## 1 Einführung

Es gibt 2 Arten von Mikroprozessoren:

- **Slice**-Prozessoren (sogenannte Bitscheiben-Prozessoren) und
- normale, wortorganisierte Mikroprozessoren auf einem **Chip** (Einchip-µP).

Sie unterscheiden sich in wesentlichen Punkten:

- *Architektur* (1)
- *Instruktionssatz* (2)
- *Wortlänge* (3)

(1) Ein Einchip-Mikroprozessor enthält alle Funktionselemente in einem Gehäuse, der Slice-Prozessor hat die Funktionselemente in verschiedene Funktionsblöcke in getrennte Gehäuse unterteilt.

(2) Der Einchip-Mikroprozessor hat einen festen Instruktionssatz, der Slice-Prozessor hat keinen Instruktionssatz im üblichen Sinn, er ist mikroprogrammiert, seine Instruktionen stehen in einem *Kontrollspeicher*. Der System-Entwickler bestimmt während des Schreibens des Mikroprogramms einen meist anwendungsbezogenen Instruktionssatz, oft auch zusätzlich einen festen, nicht von der Anwendung bestimmten Satz, der vom Benutzer nicht geändert werden kann.

(3) Ein Einchip-Mikroprozessor hat eine feste Wortlänge, z.B. 8 oder 16 bit. Beim Slice-Prozessor können mehrere Slices kaskadiert werden, daraus ergibt sich eine fast beliebige Wortlänge.

Durch diese Eigenschaften ist das Slice-Prozessorsystem meist flexibler in Architektur und Operation, es benötigt aber mehr IC-Gehäuse, mehr elektrische Leistung, höhere Kosten und längere Entwicklungs-

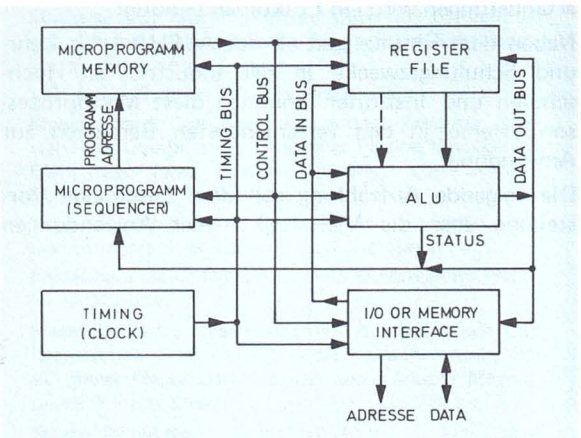



**Bild 1** Funktionsblöcke eines mikroprogrammierten Prozessors

**Slice:** (Scheibe). Hier: a) Dünne Siliziumscheibe als Grundmaterial zur Herstellung von Dioden, Transistoren oder integrierten Schaltungen; b) Prozessorelement (meist für 2 oder 4 Bit), das durch *Kaskadierung* (Zusammenschaltung) einen Prozessor größerer Wortbreite ergibt.

**Chip:** (Schnitzel, dünne Scheibe). Halbleiterscheibe mit integrierten Schaltungen.

zeit, dagegen braucht der komplette Einchip-Mikroprozessor mehr Ausführungszeit, um dieselbe Aufgabe zu lösen, d.h. Bitscheiben-Prozessoren sind schneller.

Ein typisches, verallgemeinertes Slicesystem zeigt **Bild 1.** Die abgebildeten 6 Funktionsblöcke werden im folgenden beschrieben.

## 2 Kontrollspeicher

Ein Kontroll- oder *Mikroprogrammspeicher* ist ein *N* Worte mal *M* bit Speicher, der die verschiedenen Mikroinstruktionen enthält. **Bild 2** zeigt den Speicher. Die Adressen gehen von 0 bis *N*-1. Jedes Wort enthält *M* bit, es ist in verschiedene Felder zerlegt. Im Bild sind als Beispiel 32 bit gezeichnet, Feld 1 ist 5 bit breit, Feld 2 ist 8 bit breit usw. Das Bild zeigt also das Format einer 32 bit-Mikroinstruktion.

Als Beispiel einer Aufteilung möge gelten:

Feld 1 Allgemeine Verwendung
Feld 2 **Branch**-Adresse
Feld 3 Kontrollbits der nächsten Adresse
Feld 4 Interrupt-Kontrollbits
Feld 5 Fast/Slow-Clock-Selektierung
Feld 6 **Carry**-Kontrollbits
Feld 7 ALU-**Source**-Operandenadresse
Feld 8 ALU-Funktion
Feld 9 ALU-**Destination**-Adresse

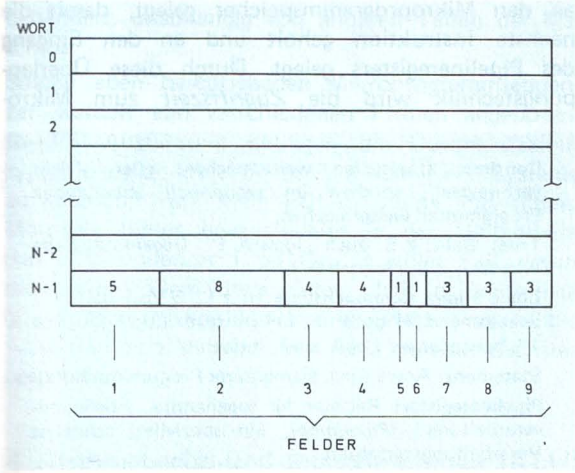



**Bild 2** Beispiel für die Aufteilung eines Mikroprogrammspeichers

## 3 Speicherkontrolleinheit (Sequenzer)

Wenn das Format festgelegt ist, muß beschrieben werden, wie die Mikroinstruktionsfolgen ablaufen sollen. Im einfachsten Fall benötigt man einen Mikroprogramm-Adreßzähler, wie es **Bild 3** zeigt.

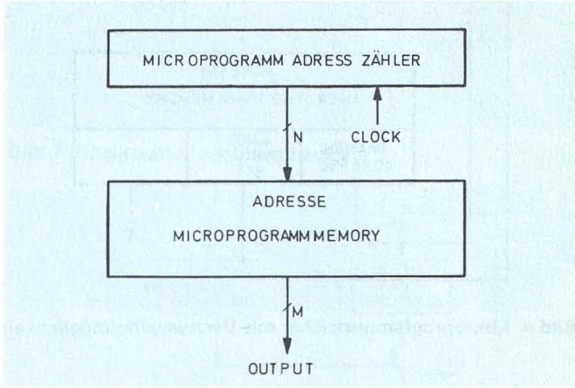



**Bild 3** Adreßzähler für den Mikroprogrammspeicher

Mit jedem Clocktakt wird der Inhalt des Zählers inkrementiert, dadurch die nächste Mikroinstruktion aufgerufen. Diese Art ist nicht sehr flexibel, das System kann nur eine feste Folge von Mikroinstruktionen ausführen. Die Inkrementschritte nennt man CONTINUE oder EXECUTE.

Wenn die Mikroprogrammeinheit auch andere als die nächste Instruktion aufrufen soll, muß sie in der Lage sein, eine BRANCH (Verzweigung)- oder JUMP (Sprung)-Adresse zu laden. **Bild 4** zeigt diese Architektur, wo eine Verzweigungsadresse parallel in den Mikroprogramm-Adreßzähler geladen wird. Die Ladekontrolle geht durch das 1 bit-Feld (*Load Enable Bit*). Ist das Bit 0, kann keine Verzweigungsadresse geladen werden, ist es 1, ist das Laden frei, und die im Mikro-

**Branch:** (Zweig; *to branch:* sich verzweigen). Die in einem Programmablauf durch logische Entscheidungen möglichen Verzweigungen.

**Carry:** (tragen, befördern). Hier: Übertrag auf nächst höhere Stelle, der bei Additionen entstehen kann.

**Source:** Quelle, Ursprung; diejenige Adresse, von der ein Datenwort zu holen ist.

**Destination:** Ziel, Bestimmung; diejenige Adresse, zu der ein Datenwort zu schaffen ist.

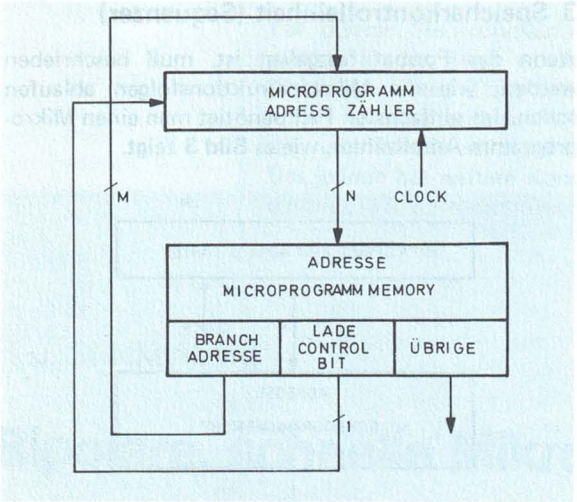



**Bild 4** Mikroprogrammspeicher mit Verzweigungsmöglichkeit

programm stehende Verzweigungsadresse wird in den Adreßzähler geladen. Dadurch wird eine *N*-fache Verzweigung möglich.

Durch die BRANCH- und JUMP-Instruktion ist die Folge von Mikroinstruktionen flexibler geworden. Was fehlt, ist noch die Möglichkeit, über Entscheidungen bedingte Verzweigungen einzuführen. **Bild 5** zeigt eine solche Kontrollanordnung für zwei verschiedene Bedingungen (*Conditions*).

Sind $S_1$, $S_0$ = 0,0, wird die Ladekontrolle inaktiv, d.h. das **CONTINUE**-Programm wird ausgeführt. Sind $S_1$, $S_0$ = 0,1 wird der D1-Eingang selektiert. Ist dabei „Cond. 1 = 0", wird der Adreßzähler inkrementiert, ist „Cond. 1 = 1", wird die *Branch-Adresse* im nächsten Clockzyklus parallel eingeladen. Ist $S_1$, $S_0$ = 1,0, wird D2 selektiert, hier gilt das gleiche. Ist $S_1$, $S_0$ = 1,1 wird D3 selektiert, d.h. **Logic High** auf "Load" gelegt. Das bedeutet, daß die Branch Adresse im nächsten Zyklus unbedingt eingeladen wird.

Dieses Beispiel zeigte die Wirkung der **Statements** CONTINUE, CONDITIONAL BRANCH, UNCONDITIONAL BRANCH.

Schauen wir uns die Konfiguration im nächsten **Bild 6** an, so sehen wir zwei Unterschiede zum vorangegangenen Bild, nämlich ein **Pipelineregister** und das Ersetzen des Adreßcounters durch eine Kombination von **Incrementer** und Register. Das Pipelineregister enthält die Mikroinstruktion, die gerade in der Maschine ausgeführt wird. Gleichzeitig aber wird die Adresse der nächsten Mikroinstruktion

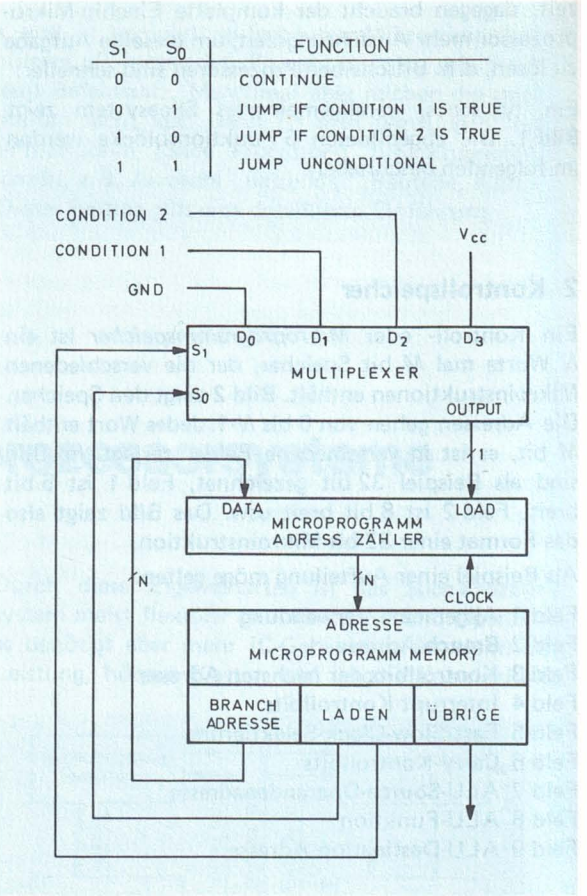



**Bild 5** Erweiterung des Mikroprogrammspeichers für bedingte Verzweigungen (abhängig von „Condition" 1 und 2)

an den Mikroprogrammspeicher gelegt, damit die nächste Instruktion geholt und an den Eingang des Pipelineregisters gelegt. Durch diese Überlappungstechnik wird die *Zugriffszeit* zum Mikro-

**Continue:** (fortsetzen, weitermachen). Hier: „Nicht verzweigen", sondern im *sequentiell* ablaufenden Programmteil weitermachen.

**True:** Wahr, z.B. auch „logisch 1". Gegensatz: *False* (Unwahr, z.B. „logisch 0").

**Logic High:** Kennbuchstabe H; z.B. 5 Volt bei TTL-Schaltungen. H bedeutet bei *positiver Logik* „logisch 1", bei *negativer Logik* aber „logisch 0".

**Statement:** Anweisung, elementarer Programmschritt.

**Pipelineregister:** Register für sogenannte „Fließbandverarbeitung" (*Pipelining*), ein spezielles, schnelles Verarbeitungsverfahren.

**Incrementer:** Einrichtung zur Erhöhung einer Adresse oder eines Wertes um eine Einheit.

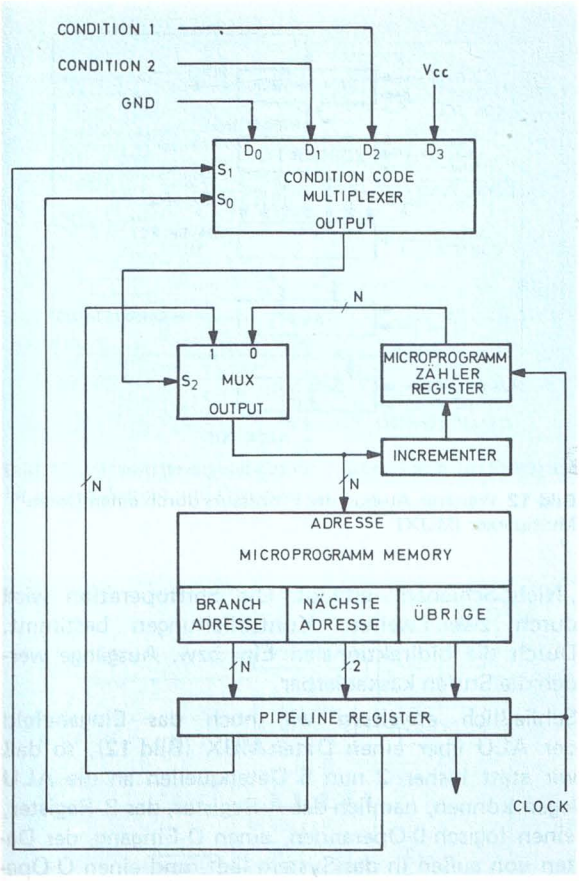



**Bild 6** Weiterer Ausbau des einfachen Mikroprogrammspeichers

programmspeicher in einer getrennten Zeitschleife behandelt, unabhängig von anderen Teilen der Maschine.

Solche eben beschriebenen Mikroprogrammsequenzer werden von verschiedenen Firmen angeboten. In TTL-Ausführung betragen die Durchlaufzeiten typischerweise etwa 50 ns. Um schnellere Systeme aufzubauen, muß man ECL-Bausteine verwenden. Motorola bietet beispielsweise in der 10800-Serie den ECL-Sequenzer 10801 an. In jedem Fall sorgen diese Kontrolleinheiten dafür, daß im Mikroprogramm die richtige Instruktion geholt und an die ALU transportiert wird. Die ALU wird als nächstes besprochen.

## 4 Arithmetische und logische Einheit (ALU)

Die erste ALU, die als Slice zu haben war, ist die bekannte 74181, die zwei Operanden je 4 bit arith-

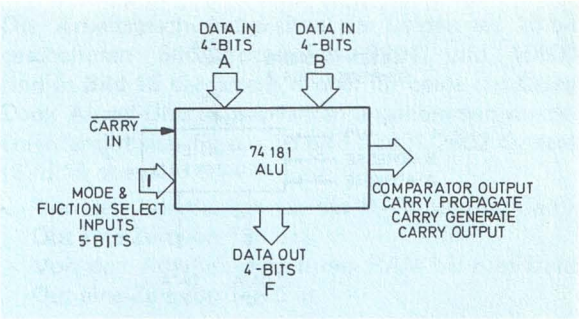



**Bild 7** Bitslice-ALU als Grundbaustein

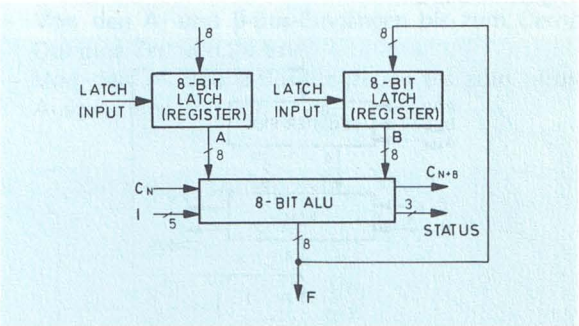



**Bild 8** ALU mit Auffangregistern (Latch)

metisch und logisch behandeln kann. Die 16 verschiedenen Funktionen werden über Mode- und Funktionsselektleitungen programmiert, die Statusein- und -ausgabesignale werden ebenso wie das Ergebnis der Operation ausgegeben. Die ALU ist in 4 bit-Schritten kaskadierbar. **Bild 7** zeigt den Baustein.

Um daraus einen Prozessor zu machen, müssen verschiedene Elemente hinzugefügt werden, z.B. die in **Bild 8** gezeigten **Latchregister,** die die Operanden halten. Der erste Operand wird in Register A geladen, die Operation F = A ausgeführt und das Ergebnis, d.h. der erste Operand, in Register B festgehalten. Der zweite Operand wird dann in Register A festgehalten, eine arithmetische oder logische Operation zwischen A und B kann ausgeführt werden. Das Ergebnis kann in Register B gespeichert oder über F ausgegeben werden.

Das Registerkonzept kann man ein bißchen ausbauen (**Bild 9**); an Stelle der Einzelregister wird ein 2-**Port**-RAM eingesetzt, zu dem 2 Adressen, nämlich A und B, gleichzeitig zugreifen können. A und B sind 4 bit-Adressen, da es 16 Register, also 16 RAM-Worte gibt.

**Latchregister:** Auffang-Flipflop, Haltekreis.

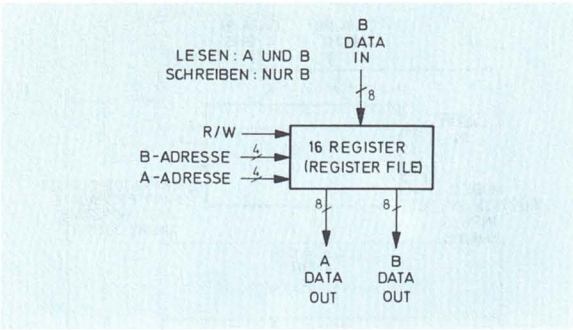



**Bild 9** Registersatz (Register File) als Erweiterung der Latch-Register in Bild 8

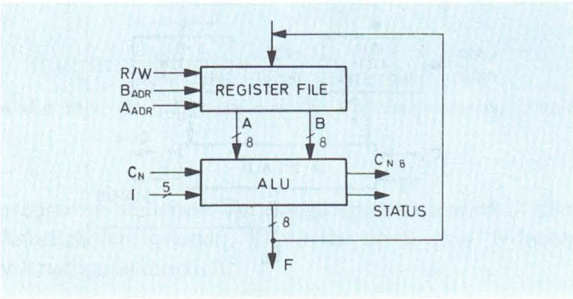



**Bild 10** Zusammenschaltung von Registersatz und ALU

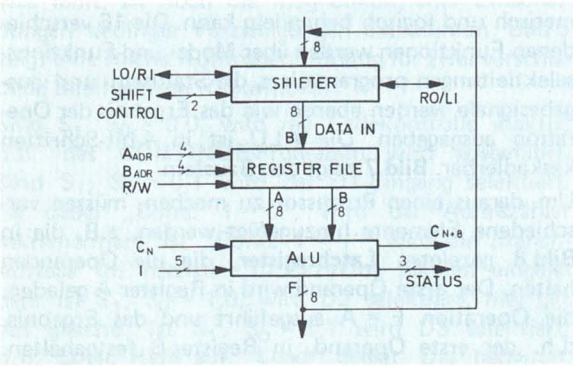



**Bild 11** Ergänzung von ALU und Registersatz durch ein „Shiftnetzwerk"

Wenn wir diesen Registersatz mit der ALU verbinden, ergibt sich **Bild 10**. Die ALU-Operanden können aus A und B gleichzeitig gelesen werden, das Ergebnis wird entweder ausgegeben oder nach B zurückgeschrieben.

**Shift** und **Rotatefunktionen** sind in einer guten CPU wichtige Operationen, deswegen werden wir unserem System ein Shiftnetzwerk zufügen (**Bild 11**), das „Links Schieben", „Rechts Schieben" und

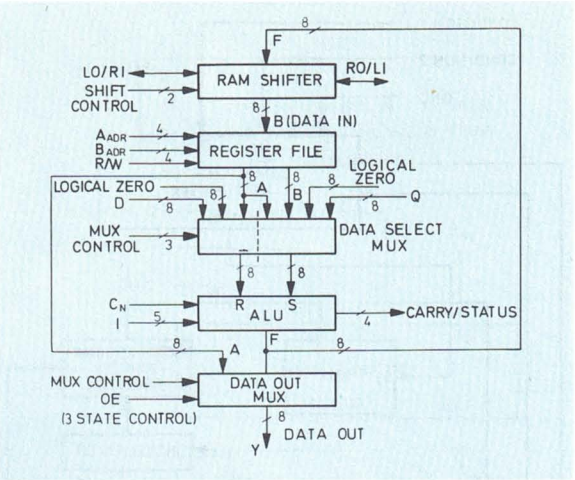



**Bild 12** Weiterer Ausbau des Prozessors durch einen Daten-Multiplexer (MUX)

„Nicht-Schieben" erlaubt. Die Shiftoperation wird durch zwei weitere Kontrolleitungen bestimmt. Durch die bidirektionalen Ein- bzw. Ausgänge werden die Stufen kaskadierbar.

Schließlich erweitern wir noch das Eingabefeld der ALU über einen Daten-**MUX** (**Bild 12**), so daß wir statt bisher 2 nun 5 Datenquellen an die ALU legen können, nämlich das A-Register, das B-Register, einen logisch-0-Operanden, einen D-Eingang, der Daten von außen in das System lädt, und einen Q-Operanden, über den wir gleich noch sprechen werden.

Die letzte Erweiterung ist in **Bild 13** zu sehen. Es wird ein Q-Register eingeführt, das für Multiplikations- und Divisionsroutinen verwendet wird und bei längeren Produkten bzw. Quotienten mit dem RAM-Register in Serie geschaltet werden kann. Dafür muß das Q-Register natürlich sein eigenes Shiftnetzwerk erhalten, das mit dem RAM-Shift in Serie geschaltet werden kann.

Nun haben wir ein System geschaffen, das der 4 bit-ALU weit überlegen ist, ein komplettes CPU-System, das in 4 bit-Slices als 2900-System von verschiedenen

**Port:** Daten-Ein- oder -Ausgang, entweder für nur ein Bit (Serienport) oder für ein ganzes Wort (Parallelport).

**Shift:** Verschiebung innerhalb eines Datenwortes um eine Einheit. „Linksschieben" bedeutet Verdopplung, „Rechtsschieben" Halbierung.

**Rotate:** Rotieren; spezielle Shift-Form, wobei das jeweils links oder rechts aus dem Datenwort geschobene Bit am anderen Ende wieder eingereiht wird.

**Mux:** Multiplexer.

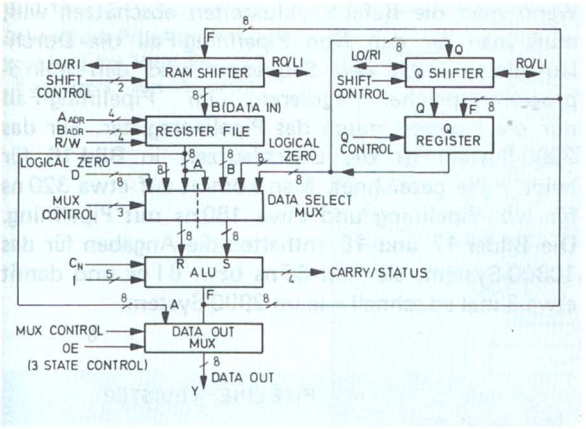



**Bild 13** Vervollständigung des Prozessors durch ein Q-Register für Multiplikation und Division

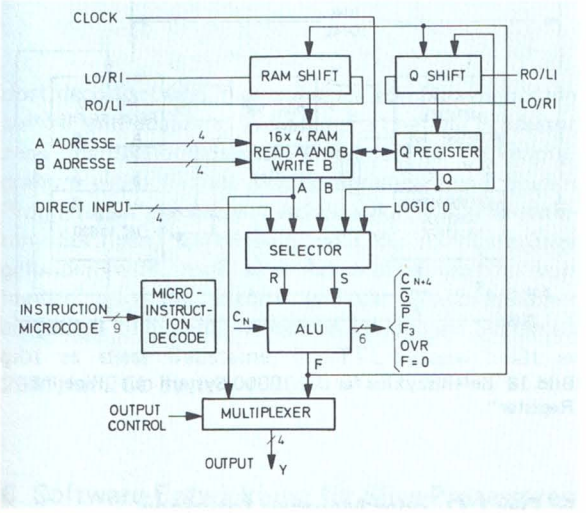



**Bild 14** Vollständiges Blockdiagramm mit dem Bitslice-Prozessor 2901

Herstellern angeboten wird (**Bild 14**). Es kann die Source-Operanden aus 5 verschiedenen Quellen holen, es kann 8 verschiedene Operationen ausführen, nämlich 3 arithmetische und 5 logische, es kann das Ergebnis entweder ausgeben oder es geshiftet oder ungeshiftet in das Q-Register oder in das RAM einladen. Das 2900-System ist in Schottky-TTL aufgebaut, es erreicht Befehlsausführungszeiten von weniger als 200 ns. Wenn wir kürzere Operationsabläufe erreichen wollen, müssen wir zur ECL-Technik übergehen. Das System 10800 ist nicht nur schneller, sondern auch leistungsfähiger und komplexer als das 2900-System.

Die Arbeitsgeschwindigkeiten der beiden als 16 bit geschalteten Slice-Prozessoren 2901 und 10800 sind in **Bild 15** dargestellt, wobei für beide das **Carry Look Ahead**-Übertragsverfahren angenommen wurde. Dann ergibt sich für ein 16 bit – 2901/2902 System (Bild 15 oben):

- Von den Adreßeingängen des RAM bis zum Carry Out eine Zeit von 120.5 ns
- Von den Adreßeingängen des RAM bis zum Data Out eine Zeit von 145.5 ns.

Für eine 16 bit – 10800/10179 System (Bild 15 unten) erhält man:

- Von den A- und Ø-Bus-Eingängen bis zum Carry Out eine Zeit von 29.5 ns
- Von den A- und Ø-Bus-Eingängen bis zum I-Bus Ausgang eine Zeit von 41 ns.

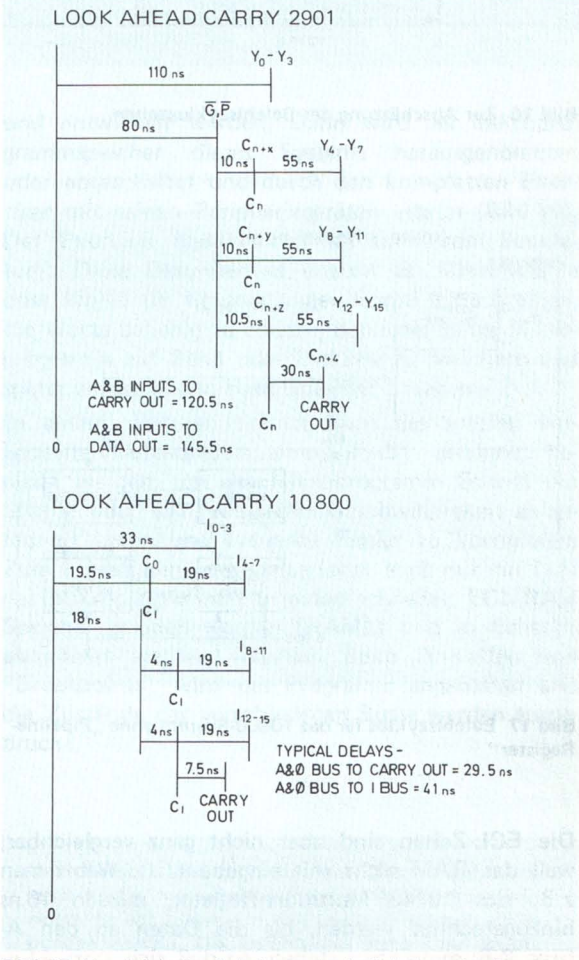



**Bild 15** Vergleich der Arbeitsgeschwindigkeiten der Prozessoren 2901 (oben) und 10800 (unten)

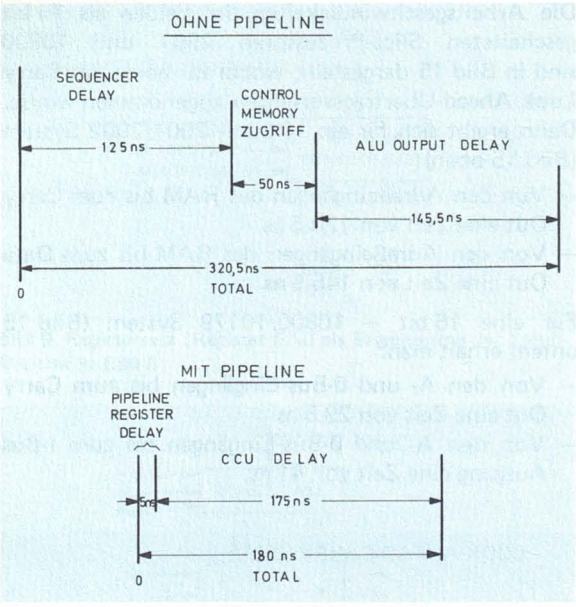



**Bild 16** Zur Abschätzung der Befehlszykluszeiten

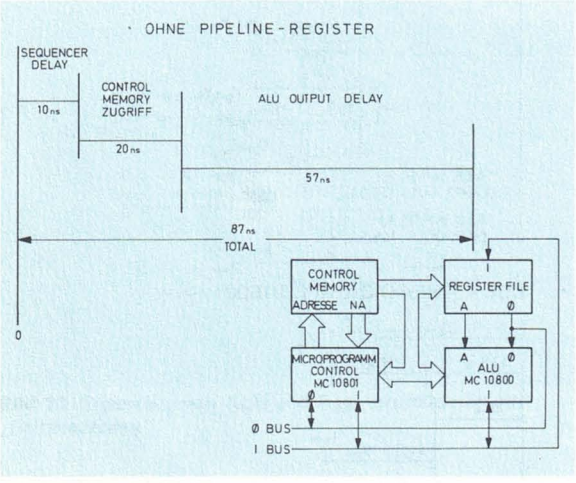



**Bild 17** Befehlszyklus für das 10800-System ohne „Pipeline-Register"

Die ECL-Zeiten sind aber nicht ganz vergleichbar, weil das RAM nicht mit eingebaut ist. Wählt man z.B. das 10143 Multiport-Register, müssen 16 ns hinzugerechnet werden, bis die Daten an den A- bzw. Ø-Buseingängen stehen, so daß sich statt 41 ns jetzt 57 ns ergeben.

Wenn man die Befehlszykluszeiten abschätzen will, muß man für den Non Pipelining-Fall die Durchlaufzeiten durch den Sequenzer und den Mikroprogrammspeicher addieren, im Pipelining-Fall nur die Laufzeit durch das Pipelineregister. Für das 2900-System ist die Durchlaufzeit in **Bild 16** für beide Fälle gezeichnet. Man kommt auf etwa 320 ns für Non-Pipelining und etwa 180 ns mit Pipelining. Die **Bilder 17 und 18** enthalten die Angaben für das 10800-System; sie sind 87 ns bzw. 61 ns und damit etwa 3 mal so schnell wie im 2900-System.

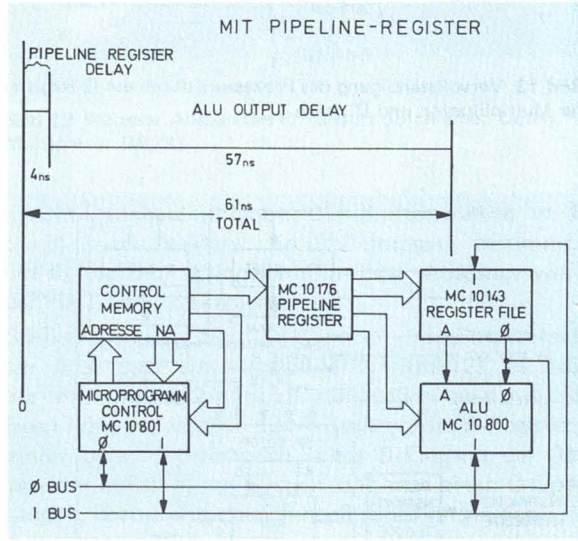



**Bild 18** Befehlszyklus für das 10800-System mit „Pipeline-Register"

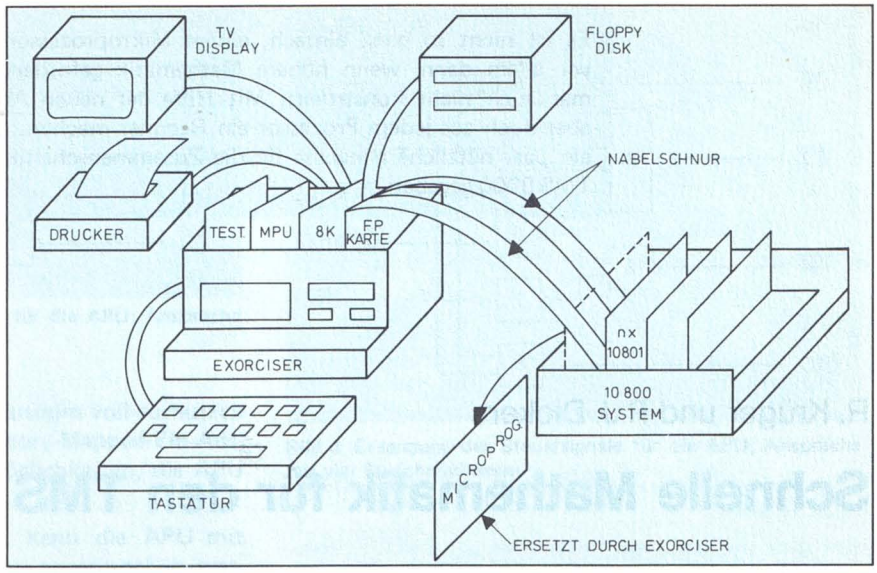
## 5 Das I/O- oder Memory Interface

Sliceprozessoren können als *Controller* oder als *Emulator* eingesetzt werden.

Als Controller enthalten sie ihr gesamtes Programm im Mikroprogrammspeicher, als Emulator haben sie im externen Speicher Instruktionen stehen, die den Maschineninstruktionen eines größeren Rechners entsprechen, z.B. einer IBM. Für den Mikroprozessor sind es **Makrobefehle,** die er nacheinander in das Instruktionsregister des Mikroprogrammsequenzers holt,

**Carry Look Ahead:** Parallel-Übertrag; spezielles Verfahren zur Erhöhung der Verarbeitungsgeschwindigkeit.

**Bild 19**
Schema der Programmentwicklung mit Hilfe des „Exorcisers" von Motorola




dort decodiert und in eine Anzahl von Mikroschritten auflöst und ausführt. In diesem Fall hat der Prozessor zwei Instruktionsspeicher, einen für das Makroprogramm, einen für das Mikroprogramm. Da in beiden Programmen die nächste Adresse durch Inkrementieren, Springen, Verzweigen oder Subroutinesprünge gefunden wird, muß eine Art *Makrosequenzer* vorhanden sein, der ähnlich wie der *Mikrosequenzer* aufgebaut sein wird. In beiden erwähnten Systemen gibt es diese Bausteine, im TTL-System heißt er 2930, im ECL-System 10803.

## 6 Software-Entwicklung für Slice-Prozessoren

Die Auslegung von verschiedenen Benutzersystemen variiert sehr stark je nach der Anwendung (Controller, Emulator etc.) Aus diesem Grunde ist es sehr schwierig, einen allgemeinen Assembler, Editor und Simulator für alle möglichen Anwendungen zu konzipieren. Andererseits ist es sehr mühsam, wenn nicht gar unmöglich, ein Mikroprogramm eines gewissen Umfanges ohne jedes Hilfsmittel zu entwickeln und dessen Fehler zu bereinigen. *Motorola* hat deshalb ein spezielles Systementwicklungsgerät herausgebracht, das erstens das Schreiben von Mikroprogrammen und zweitens die Fehlerbereinigung dieses Mikroprogrammes ganz wesentlich erleichtert.

Um das Systementwicklungsgerät einsetzen zu können, muß zuerst das Benutzersystem konzipiert und entwickelt werden. Dann wird der Mikroprogrammspeicher dieses Systems herausgenommen oder abgeschaltet und durch den kompletten *Exorciser* mit seinen Peripheriegeräten ersetzt (**Bild 19**). Der Exorciser figuriert nun als ein "Prom Simulator". Diese Besonderheit erlaubt es, Mikrobefehle oder Worte (in hexadezimaler Form) zu schreiben, die Worte beliebig zu ändern, dann das ganze Mikroprogramm auf Band oder Diskette zu speichern und später wieder in den Hauptspeicher zu laden.

In einem weiteren Schritt kann das speziell entwickelte Softwareprogramm, FAST genannt, benutzt werden, um das Mikroprogramm Schritt um Schritt oder mit reduzierter Geschwindigkeit ablaufen zu lassen und eventuell Fehler zu korrigieren. Zum Schluß kann das ganze (oder auch nur ein Teil) des Mikroprogrammes in einen schnellen ECL-RAM Speicher geladen werden (RAML) und in Echtzeit ausgeführt werden (REAL). Beim Antreffen von "Breakpoints" wird das Programm angehalten und die Zustände der verschiedenen Busse werden ausgedruckt.

**Makrobefehl:** (*Macro Instruction*). Zusammenfassung einer häufig wiederkehrenden Befehlsfolge kann mit einem Befehl aufgerufen und in das Programm eingebunden werden. Die Befehlsfolge steht dann so oft im übersetzten Programm, wie sie aufgerufen wird.

Es ist nicht so ganz einfach, einem Mikroprozessor das Rechnen beizubringen, vor allem dann, wenn höhere Mathematik gefordert wird. Dafür sind ja µP primär auch nicht konstruiert. Mit Hilfe der neuen Arithmetik-Bausteine läßt sich aber doch aus jedem Prozessor ein Rechner machen. Im folgenden Aufsatz werden ein paar nützliche Hinweise für die Zusammenschaltung des AM 9511 mit dem µP TMS 9900 gegeben.

R. Krüger und P. J. Dickers

# Schnelle Mathematik für den TMS 9900

In diesem Beitrag wird der Anschluß des schnellen Arithmetik-Prozessors AM 9511 an den Mikroprozessor TMS 9900 beschrieben. Der AM 9511 von AMD ist eine *8 bit Arithmetic Processing Unit* (**APU**) mit typischen Befehlsausführungszeiten von 49 µs für eine Gleitkomma-Multiplikation und 1,3 ms für die Berechnung eines Sinusfunktionswertes. Es können sowohl **Gleitkomma-** als auch **Festkomma-Operationen** ausgeführt werden. Der Baustein arbeitet intern mit einem Stackregister und mit UPN (Umgekehrte Polnische Notation).

Von den drei Geschwindigkeitsversionen wurde hier die 3-MHz-Ausführung verwendet, passend zum Systemtakt des Mikroprozessors. Gefordert ist für die APU ein einfacher unsynchronisierter Takt, dessen Tastverhältnis so zu wählen ist, daß der **High-Pegel** mindestens 140 ns und der **Low-Pegel** mindestens 160 ns lang sind. Von dem Vierphasentakt des Mikroprozessors werden auf der TMS 990/100-Platine die $\overline{\Phi}_1$- und $\overline{\Phi}_3$-Signale herausgeführt. Aus diesen beiden Signalen läßt sich das geforderte Taktsignal für die APU leicht gewinnen (siehe **Bild 1**).

Befehle, Statusausgaben und Daten werden über denselben **I/O-Bus** der APU übertragen. Zur Steuerung werden vier Signale benötigt:

1. $\overline{CS}$ **Chip-Select**
2. C/$\overline{D}$ Control/Data: High = Befehl oder Statusabfrage
   Low = Data
3. $\overline{RD}$ Lesebefehl
4. $\overline{WR}$ Schreibbefehl

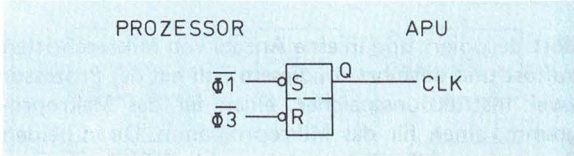



**Bild 1** Erzeugung des Taktsignals CLK für die APU aus den Taktsignalen $\overline{\phi}_1$ und $\overline{\phi}_3$ des Mikroprozessors (CLK: Clock)

**APU:** *Arithmetic Processing Unit.* Rechenwerk, das aus einem µP einen technisch-wissenschaftlichen Rechner macht.

**Gleitkomma-Arithmetik:** Das auch bei technisch-wissenschaftlichen Taschenrechnern benutzte Verfahren zur Darstellung und Verarbeitung sehr großer Zahlen, wobei das Komma sich immer automatisch hinter die erste Stelle setzt und die Stellenzahl als Zehnerpotenz angegeben wird.

**Festkomma-Arithmetik:** Verarbeitung mit Festlegung auf z. B. 2 oder 4 Nachkommastellen.

**High-Pegel:** (H oder HI). Damit ist der auf die zweite *Binärvariable* bezogene höhere Potentialzustand gemeint (1 bei positiver Logik oder 0 bei negativer Logik).

**Low-Pegel:** (L oder LO). Damit ist der auf die zweite Binärvariable bezogene niedrigere Potentialzustand gemeint (0 bei positiver Logik oder 1 bei negativer Logik).

**CLK:** (*Clock*). Taktgeber

Φ: Abkürzung für die Phase des Taktgebers. Darüberliegender Querstrich bedeutet Phasenumkehr (Verneinung).

**I/O-Bus:** Ein-Ausgabe-Bus (*Input/Output*).

**Chip-Select:** Chip-Auswahl-Leitung.

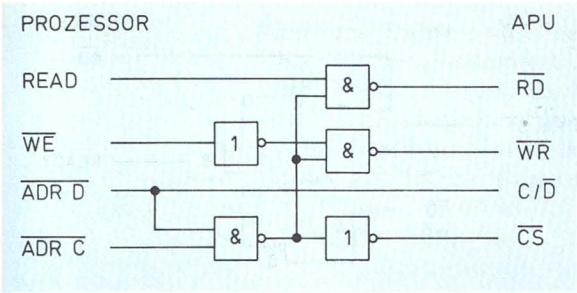



**Bild 2** Erzeugung der Steuersignale für die APU; Ansprache mit zwei Speicheradressen

Um die Geschwindigkeit des Bausteins voll ausnutzen zu können, bietet sich die **Memory-Mapped-Ein-Ausgabe** an. Es gibt dann zwei Möglichkeiten, die APU anzusteuern:

Wie aus **Bild 2** deutlich wird, kann die APU mit zwei einzelnen Speicheradressen angesprochen werden. Die eine, ADR C, wird nur dann angesteuert, wenn ein Befehl eingelesen oder eine **Statusangabe** ausgelesen wird. Die andere, ADR D, wird für den Datentransfer benötigt. Diese Adressierung wird für die internen Ports der APU gebraucht. Die decodierte Einzeladreßleitung ADR D dient dabei gleichzeitig als C/D-Signal. Der Schreibimpuls WE des Mikroprozessors gibt mit dem CS-Signal verknüpft das WR-Signal. Das **READ** ist eine Einzelbitleitung der CRU (*Communication Register Unit* des TMS 9900).

Zum Einschreiben von Befehlen und Daten in die APU muß diese Leitung *High,* zum Auslesen *Low* gesetzt werden. Da der TMS 9900 auch vor jeder Schreiboperation liest und die APU dabei ihre Daten verliert, ist es unmöglich mehr als 8 Bit einzuschreiben, wenn das Lesesignal des Mikroprozessors DBIN (*Data Bus In*) als READ-Signal verwendet wird. In diesem Fall würden beim nächsten Schreibbefehl durch das Lesen vor dem Schreiben die zuletzt eingegebenen 8 Bit aus der APU verloren gehen.

Die zweite Möglichkeit ist die in **Bild 3** dargestellte Verwendung von vier Speicheradressen, so daß es sowohl getrennte Adressen für Kontrolle und Daten als auch für Schreiben und Lesen gibt. Die zusätzliche Verknüpfung mit dem WE-Signal ist wieder notwendig, weil der Mikroprozessor vor dem Schreiben liest. Ohne diese Verknüpfung würde die Information (Data oder Befehl) zweimal eingeschrieben.

Für das *Handshaking* mit dem Mikroprozessor stellt die APU zwei Leitungen END und PAUSE zur Verfügung.

Das END-Signal gibt das Ende einer Operation an.

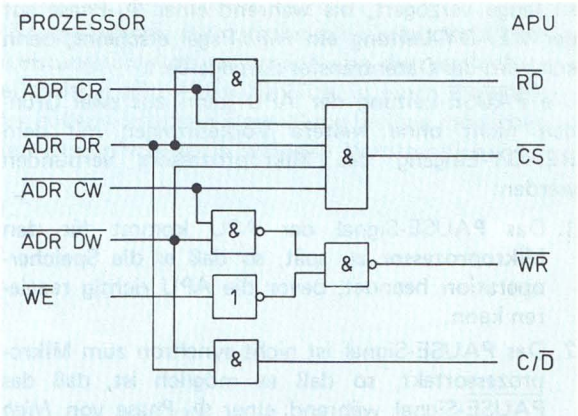



**Bild 3** Erzeugung der Steuersignale für die APU; Ansprache mit vier Speicheradressen

Diese **Open-Drain-Leitung** kann, wenn gefordert, direkt an einen Interrupt-Eingang des Mikroprozessors gelegt werden.

Das PAUSE-Signal wird aktiv:

1. wenn die APU angesprochen wird und zu diesem Zeitpunkt der letzte Befehl noch nicht abgearbeitet ist;
2. so lange, bis ein eingegebener Befehl erfaßt ist;
3. bei einem Datentransfer so lange, bis entweder die Daten richtig erfaßt sind oder gültige Daten am Ausgang stehen.

Der TMS 9900 besitzt einen READY-Eingang. Dieser wird bei einem normalerweise zwei Taktzyklen dauernden Speicherzyklus während der ersten $\Phi_1$-Phase abgefragt.

Ist dabei das READY-Signal *High,* wird der Datentransfer während des nächsten Taktzyklus durchgeführt und die Speicheroperation beendet. Ist das READY-Signal *Low* wird die Speicheroperation

**Memory-Mapped-Ein-Ausgabe:** Ein-Ausgabe mit festgelegter Speicherbereichszuweisung.

**Status:** Die Gesamtheit der nach Ausführung jedes Befehls entstandenen logischen Zustände. Sie werden als sogenannte Statusbit (*Flags*) abgespeichert.

**WRITE:** Schreibsignal

**READ:** Lesesignal

**Open-Drain-Leitung:** Leitung an „offenem" Drain-Anschluß eines Feldeffekttransistors. Vergleichbar mit *Open Collector* (offener Kollektor) bei bipolaren Schaltungen.

so lange verzögert, bis während einer $\Phi_1$-Phase auf der READY-Leitung ein *High*-Pegel erscheint, dann erst wird der Datentransfer durchgeführt.

Die $\overline{\text{PAUSE}}$-Leitung der APU kann aus zwei Gründen nicht ohne weitere Vorkehrungen mit dem READY-Eingang des Mikroprozessors verbunden werden:

1. Das PAUSE-Signal der APU kommt für den Mikroprozessor zu spät, so daß er die Speicheroperation beendet, bevor die APU richtig reagieren kann.
2. Das PAUSE-Signal ist nicht synchron zum Mikroprozessortakt, so daß es möglich ist, daß das $\overline{\text{PAUSE}}$-Signal während einer $\Phi_1$-Phase von *High* nach *Low* geht.

Letzteres führt beim TMS 9900 zu unkontrollierbaren Reaktionen.

Um dem zu begegnen, kann man das $\overline{\text{PAUSE}}$-Signal mit dem $\overline{\Phi_3}$-Signal synchronisieren. Weiterhin sorgt eine Verknüpfung der $\overline{\text{RD}}$- und $\overline{\text{WR}}$-Signale in Verbindung mit einem Zeitglied von ca. 150 ns für eine rechtzeitige Auslösung eines Wartezyklus beim Mikroprozessor (siehe **Bild 4**).

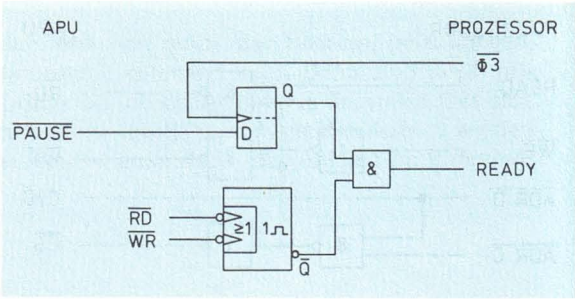



**Bild 4** Synchrone Erzeugung von Wartezyklen für den Prozessor

Bei der Berechnung von Programmausführungszeiten sind zwei Dinge zu berücksichtigen:

1. Einerseits addieren sich zu den Befehlsausführungszeiten der APU die Zeiten für die Ein- und Ausgaben.
2. Andererseits arbeiten Prozessor und APU (auch ohne Interrupt-Steuerung) teilweise parallel.

Daher ist es sehr programmabhängig, wie sich die Befehlsausführungszeiten der APU au die Programmausführungszeit auswirken.

**Angewandte Informatik**
**applied informatics**

Diese Zeitschrift bringt Berichte über die Situation auf dem Gebiet der Büro-Automation und des elektronischen Rechners, über Betriebsautomatisierung, über Möglichkeiten, Einsatz und Installierung neuer Rechenanlagen, über die Praxis der Programmierung mit konkreten Beispielen, über Neuentwicklungen der Grundlagenforschung. Ferner Tagungs-, Ausstellungsberichte und Buchbesprechungen. Zusammenfassungen in deutscher und englischer Sprache.

**Datenschutz und Datensicherung**
zugleich „Der Datenschutzbeauftragte"

**Datenschutz und Datensicherung** behandelt die praktische Auslegung der gesetzlichen Bestimmungen, die Maßnahmen zur Befolgung des BDSG und die damit verbundenen Probleme in Kurzmitteilungen, praxisnahen Originalbeiträgen und Erfahrungsberichten.
Das integrierte Informationsorgan **Der Datenschutzbeauftragte** bringt aktuelle und gesicherte Informationen: Hinweise, Ratschläge, Kurzmitteilungen, neue Rechtsvorschriften, Rechtsprechung etc.

**Fragen Sie Ihren Buchhändler nach einem Probeheft oder schreiben Sie an: Verlag Vieweg, Pf. 5829, 6200 Wiesbaden 1.**

Bei der Aufwertung eines Mikroprozessors zum technisch-wissenschaftlichen Rechner hängen Komfort und Leistungsfähigkeit wesentlich von der geschickten Programmorganisation ab. Dieser Beitrag gibt einen Einblick in einen eleganten Mathematik-Assembler. Wegen der äußerst knappen Darstellung bleiben möglicherweise Fragen unbeantwortet. Die Autoren werden in weiteren Veröffentlichungen darauf ausführlicher eingehen.

R. Krüger und P. J. Dickers

# Mathematik Assembler für Mikroprozessoren

Die Assemblersprachen von Mikroprozessoren eignen sich vorzüglich für Steuerungszwecke und zum Manipulieren von Daten, sind jedoch recht umständlich, wenn höhere mathematische Operationen bearbeitet werden sollen.

Der vorliegende Artikel beschreibt eine Methode, wie eine Reihe von fertigen **Mathematikroutinen** zu einer einfachen Assemblersprache zusammengefaßt werden kann. Diese Mathematikroutinen können entweder aus einer Folge von $\mu$P-Befehlen bestehen oder von einem fest verdrahteten Rechenwerk ausgeführt werden. Die Sprache besteht aus ein- bis dreibuchstabigen Befehlen, denen ggf. weitere Angaben, Adressen oder Daten folgen können.

Es wird Umgekehrte Polnische Notation (UPN) und ein Stack-Register verwendet. Dies erspart die Angabe von **Operanden** im Befehl, da Operationen immer auf die obersten beiden Register im Stack wirken.

Der Mathematik-Befehlsablauf steht in der Form von aufeinanderfolgenden **Operationscodes** wie ein Unterprogramm an irgendeiner Stelle im RAM. An der Stelle im Assembler-Hauptprogramm, an der eine oder mehrere mathematische Operationen benötigt werden, steht jetzt ein Unterprogrammsprung. Es wird ein Steuerunterprogramm aufgerufen, das jedes Mathematik-Befehlszeichen (*Op-Code*) liest, interpretiert und den Sprung in ein weiteres Unterprogramm veranlaßt, das die entsprechende Funktion ausführt. Nach Rückkehr aus diesem Unterprogramm wird die nächste Mathematik-Befehlszeile vom Steuerunterprogramm abgefragt und entsprechend abgearbeitet, solange bis END erscheint. Dann erfolgt der Rücksprung ins Hauptprogramm.

Der Vorteil dieser Organisationsstruktur liegt für den Anwender darin, daß es nicht nötig ist, eine Vielzahl von Unterprogrammadressen vom Hauptprogramm aus aufzurufen und diese geeignet miteinander zu verbinden. Es genügt die Übergabe der Startadresse der Mathematik-Operations-Codes, die ausgeführt werden sollen, und zusätzlich der Sprung in das Steuerunterprogramm.

Beim System TMS 990/100 von Texas Instruments läßt sich beides in einer „Erweiterten Operation" angeben, z.B.

XOP @ MATHE, 5

**Routine:** Selbständiger Programmteil, der beliebig in Hauptprogramme gerufen werden kann (auch: *Subroutine* oder Unterprogramm).

**Operand:** Meßwert oder andere im Rechner gespeicherte Information, mit oder an der eine dem Befehl entsprechende *Operation* ausgeführt werden soll.

**Operationscode:** Programmbefehle bestehen normalerweise aus *Adreßteil* und *Operationsteil*. Der Adreßteil weist auf den Operanden, der Operationscode (auch *Opcode* oder Befehlscode) gibt im Operationsteil an, was zu geschehen hat.

**Hauptprogramm**

```
6000  LWPI >FE00          Arbeitsregistersatz laden
6004  LI R0 >0004 = 4
6008  BL @>602E           Data erfassen von Gerät 4
600C  MOV R5,@>7C06
6010  CLR @>7C04          Data übergeben an Register 1
6014  LI R0 >0006 = 6
6018  BL @>602E           Data erfassen von Gerät 6
601C  MOV R5,@>7C0A
6020  CLR @>7C08          Data übergeben an Register 2
6024  XOP @>7000,5        Mathematik-Unterprogramm aufrufen
6028  B @>0080            Rücksprung zum Monitor
602C  ENDE
```

**Mathematik-Unterprogramm für sin ($R_1$ + $R_2$)**

```
7000 5801 RCL 1           Register 1 aufrufen
7002 0200 FAC             Umwandeln auf Gleitkomma
7004 5802 RCL 2
7006 0200 FAC
7008 1800 +
700A 3000 SIN
700C 2000 SQR
700E 6E00 FRT 6100        Unterprogramm TEXTAUSG.
7010 6100
7012 5005 FIX 5           Ausgabeformat
7014 4A00 OUX             X-Register ausgeben
7016 0000 END             Rücksprung ins Hauptprogramm
```

**Unterprogramm TEXTAUSG.**

```
6100  XOP @>6108,14       XOP 14 : UP Textausgabe
TEXT:
ERGEBNIS =
6104  B *R11              Rücksprung in das Mathematik
6106  ENDE                Unterprogramm
```

**Ausführung des Programms**

```
*?R 6000

ERGEBNIS = .62923
*?
```

**Bild 1** Programmbeispiel mit dem Mikroprozessor TMS 9900

Mit diesem Assembler-Befehl wird das Steuerunterprogramm aufgerufen. Dieses führt die Befehle aus, deren Operationscodes mit der angegebenen Adresse (hier MATHE) beginnen (siehe auch **Bild 1**).

Zur Übersetzung erstellt man eine Tabelle aller mnemonischen Befehle, die maximal aus je drei Buchstaben und einem Beendigungszeichen (Null-Byte) bestehen. Hat der Befehl weniger Buchstaben, wird der Platz durch Beendigungszeichen aufgefüllt. Jeder Befehl benötigt 4 Byte. Jeder eingelesene Befehl wird vom **Übersetzer-Programm** in der Tabelle gesucht. Von der Adresse des gefundenen Befehls wird die Startadresse der Tabelle subtrahiert und durch 2 dividiert. Das Ergebnis ist der binäre Befehlscode. In dem Beispiel **Bild 1** ist im Mathematik-

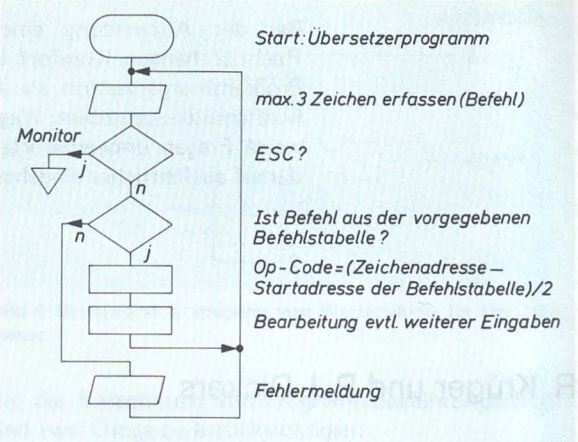



**Bild 2** Schematisiertes Flußdiagramm für die Übersetzung (**ESC:** Steuertaste auf einer **ASCII-Tastatur**)

Unterprogramm der Operationscode in der zweiten Spalte ausgedruckt. **Bild 2** zeigt das Flußdiagramm für die Übersetzung.

Die Reihenfolge der Befehle in der Tabelle sollte davon abhängen, ob der Befehl ein zweites, drittes oder mehr Byte benötigt. Der so erhaltene binäre Befehlscode wird mit einer **Befehlsformat-Tabelle** verglichen, in der die Adressen aufgeführt sind, bei denen Ein-, Zwei- oder Mehrbyte-Befehle beginnen oder bei denen Befehle beginnen, die gesondert behandelt werden müssen. Dies ist beispielsweise die Eingabe von Konstanten. Entsprechend dem Ergebnis dieses Vergleiches mit der Befehlsformat-Tabelle springt dann das Übersetzerprogramm in einen Programmteil, in dem die noch nachzutragenden Daten übersetzt und auf die Adressen geschrieben werden, die dem Operationscode folgen.

**Übersetzer-Programm:** Alle in Assembler- oder höheren Sprachen geschriebenen Programme müssen vor dem Start in den Maschinencode übersetzt werden. Dazu dienen Übersetzer-Programme: *Assembler, Compiler* oder *Interpreter.*

**ESC:** *Escape* (Flucht). Eine der ASCII-Steuertasten. Sie bewirkt eine Veränderung der Bedeutung des nächstfolgenden Zeichens gegenüber der in der Code-Tabelle angegebenen (Umschaltung).

**ASCII-Tastatur:** Weitgehend genormte Tastatur von Fernschreibern (TTY), Bildschirmterminals etc. zur Dateneingabe im ASCII-Code. Anordnung etwa wie bei Schreibmaschine.

**Befehlsformat:** Gibt erstens an, wieviele Byte zum Befehl gehören (Befehlslänge) und legt zweitens die Länge von Operations- und Adreßteil fest.

Bei der Interpretation dieser Operationscodes durch das XOP n-Unterprogramm genügt es, wenn der Code zur Startadresse einer Sprungtabelle addiert wird (siehe **Bild 3**).

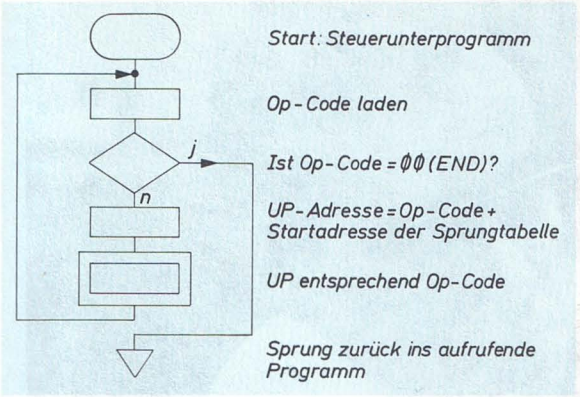



**Bild 3** Schematisiertes Flußdiagramm für das Steuerunterprogramm

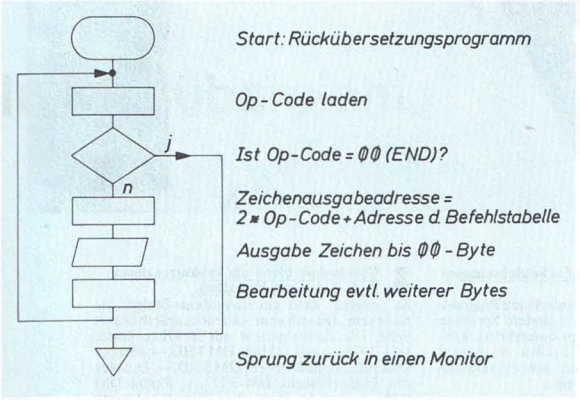



**Bild 4** Schematisiertes Flußdiagramm für die Rückübersetzung

Es werden nur geradzahlige Operationscodes verwendet, um 16 Bit-Adressen in der Sprungtabelle ansteuern zu können. Diese Adressen sind die Startadressen der einzelnen Mathematik-Unterprogramme. Für das END wird die erste Stelle der Befehlstabelle also das Null-Byte als Operationscode gewählt. Dieser dient dem Steuerprogramm als Signal, in das Hauptprogramm zurückzugehen.

Die Rückübersetzung erfolgt ähnlich. Der Code wird mit 2 multipliziert, anschließend wird die Startadresse der Befehlstabelle, die der Übersetzung diente, addiert und der Inhalt bis zum Beendigungszeichen ausgelesen (siehe **Bild 4**). Anschließend wird der Operationscode mit der Befehlsformat-Tabelle verglichen, um zu erfahren, wieviel Byte diesem Befehl folgen und wo diese Byte zurückübersetzt werden.

Innerhalb einer Folge von Mathematik-Assembler-Befehlen kann es nötig sein, in den Prozessor-Assembler zurückzukehren. Dies ist zum Beispiel nützlich zur **Formatierung** von Ausgaben oder Textausdrucken. Deshalb wird ein entsprechender Unterprogrammsprung (hier FKT genannt) in den Befehlssatz aufgenommen (siehe Bild 1).

Ein vollständiges Programm dieser Art für das System TMS 990/100 unter Verwendung des schnellen Arithmetik-Prozessors AM 9511 soll 1979 in einem PTB-Bericht veröffentlicht werden.

**Formatierung:** Sortieren und Anordnen von Befehlen und Daten in festgelegten Formen, Abständen etc. Wichtig z.B. auch bei Abspeicherung auf Magnetband oder Floppy-Disk.

# „Natürlich propagieren viele den Computer am Arbeitsplatz. Doch was bringt mir das?"

Professional Partners

## Commodore weist die Richtung:

**1. Commodore macht Sie von der zentralen Rechenanlage unabhängig.**
Das gilt uneingeschränkt für alle Daten, die nicht für den täglichen betrieblichen Arbeitsfluß notwendig sind und deshalb auch nicht zentral erfaßt und verarbeitet werden müssen. Mit dem neuen Microcomputer-System cbm 3001 haben Sie jederzeit den Zugriff auf spezielle, allein auf Ihren Arbeitsplatz bezogene Programme. Die Peripherie des cbm 3001 besteht ausschließlich aus intelligenten Geräten, sodaß Sie immer über die volle Speicherkapazität verfügen können.

**2. Commodore entlastet die betriebseigenen EDV-Spezialisten.**
cbm 3001 arbeitet mit der einfachsten Programmiersprache der Welt: Basic (andere Sprachen werden vom Betriebssystem unterstützt). Kleine Programme schreiben Sie selbst. Auch sonst können Sie jedes Programm schnell variieren und somit immer aktualisieren.

**3. Commodore bietet ein konkurrenzloses Preis-Leistungs-Verhältnis.**
So „schmal" kann ein Investitions-Budget gar nicht sein, daß sich cbm 3001 nicht unterbringen ließe. Die Zentraleinheit mit 16 kByte kostet DM 2938,--, mit 32 kByte DM 3503,--, Floppy-Disk mit Doppellaufwerk DM 3503,--, Drucker mit Traktorführung DM 3277,--, Plotter DM 7485,-- und der inzwischen überall bekannte Personal Electronic Transactor 2001 mit 4 kByte sogar nur DM 1895,-- (jeweils unverbindliche Preisempfehlung, inkl. 13% MwSt.)

Commodore ist ein international führender Electronic-Hersteller, der mit eigener Microprozessor-Technologie dieses bisher einmalige Verhältnis zwischen Preis und Leistung bietet.

**cbm 3001 Serie**
**DM 3.503,-**
(unverbindliche Preisempfehlung für 32 kByte Zentraleinheit)

**Informations-Gutschein**
für kostenlose Übersendung von Prospektmaterial

Name: _____

Anschrift: _____

_____

_____

Einsenden an: Commodore GmbH
Frankfurter Str. 171–175 · 6078 Neu-Isenburg

C

**commodore**

Commodore GmbH · Frankfurter Str. 171–175
6078 Neu-Isenburg · Tel. (06102) 8003 · Telex: 4185663 como d

# Speicher

## Inhaltsübersicht

Einführung . . . . . 154

*Harald Schumny*
Aufzeichnungsverfahren für magnetische Datenträger . . . . . 155

*Harald Schumny*
Beispiele für die Übertragung, Speicherung und Sicherung von Daten . . . . . 160

*O. Neufang*
Halbleiterspeicher . . . . . 166

*Wolfgang Link*
Programmierbare Festwertspeicher (PROM) . . . . . 171

*Harald Schumny*
Bipolare Speicher-Technologien . . . . . 177

*Wolfgang Link*
CCD-Halbleiterspeicher ersetzen rotierende Massenspeicher . . . . . 181

# Einführung

Speicher bilden das „Gehirn" einer Datenverarbeitungsmaschine. Ohne Speicher ist solch eine Maschine nichtmal zur Ausführung eines einfachsten Schrittes in der Lage; denn jeder Verarbeitungsschritt wird erst durch *Mikroinstruktionen* möglich, die vom Hersteller in Festwertspeichern abgelegt sind. Mikroprozessoren haben außerdem einzelne Speicherstellen fest „eingebaut", z.B. Akkumulator oder Programmzähler. Für Programme und Daten benötigt man Schreib-Lesespeicher und oft weitere Festwertspeicher. Für große Datenmengen schließlich braucht man einen preiswerten Massenspeicher, z.B. eine Magnetbandkassette.

Aus diesen Beispielen wird deutlich, daß es für verschiedene Zwecke unterschiedliche Speichertypen gibt – unterschiedlich in der Art des Zugriffs, in der Zugriffsgeschwindigkeit, der Speicherkapazität und im Preis. Eine Einteilung kann etwa folgendermaßen aussehen:

- *Register* (einzelne Speicherstellen) für die kurzfristige Speicherung von Parametern, Zwischenergebnissen, logischen Zuständen, Ein-Ausgabeinformationen;
- *Festwertspeicher* (ROM) zur Aufnahme von Betriebsprogrammen (Monitor, Assembler etc.), von fertigen Anwenderprogrammen und anderen, unveränderlichen Angaben;
- *Schreib-Lesespeicher* (RAM) zur Aufnahme veränderlicher Daten und Operanden und als Arbeitsspeicher für die Programmentwicklung und den Verarbeitungsablauf;
- *Massenspeicher* (Kassette, Floppy, Platte) zur Zwischenspeicherung langer Programme und großer Datenmengen und zur langfristigen Archivierung.

Register und Speicher vom Typ ROM und RAM sind heute als Halbleiterspeicher ausgeführt und schon erstaunlich preiswert. „Normale" RAM jedoch sind flüchtig, d.h. sie verlieren beim Abschalten der Versorgungsspannung ihre Information. Die Nutzung löschbarer und wiederprogrammierbarer Festwertspeicher (EPROM) setzt das Vorhandensein eines „Programmers" und einer Löscheinrichtung voraus; beides zusammen kostet in der Regel erheblich mehr als ein Heimcomputer.

Diese Überlegungen lassen verstehen, warum sich die „Musikkassette" so stark durchgesetzt hat: Laufwerke und Kassetten sind sehr billig, die Speicherung ist dauerhaft und trotzdem löschbar, nicht zu lange Programme sind in vertretbarer Zeit abgespeichert bzw. in den Computer zurückgeladen.

Einen eindrucksvollen Kostenvorteil gewinnt man aber nur, wenn einfache Aufzeichnungsverfahren verwendet werden und – siehe oben – billige Musikkassetten und Recorder benutzt werden können. Dieser Weg wird von den meisten Lern- und Heimcomputer-Herstellern beschritten. Man bekommt z.B. den KIM mit Kassettenrecorder-Interface, den TRS 80 mit getrennt aufstellbarem Recorder, den PET mit eingebautem Laufwerk. Zur Aufzeichnung der Bitfolgen werden hauptsächlich Verfahren der Frequenzcodierung genutzt, wobei *Frequenzumtastungen* im Vordergrund stehen. In diesen Zusammenhang gehören Bezeichnungen wie *KIM-Format* oder *Kansas-City-Standard.* Der erste Beitrag „Aufzeichnungsverfahren für magnetische Datenträger" wird hierauf näher eingehen.

Dem Kostenvorteil steht gegenüber, daß die Schreib-Lesegeschwindigkeit bei gewöhnlichen Musikrecordern sehr klein ist. Und selbst für denjenigen, der viel Zeit mitbringt, hört dann der Spaß auf, wenn seine preiswerte Kombination aus „Primitivrecorder" und „Kaufhauskassette" ständig zu Schreib-Lesefehlern führt. Dann bieten sich zwei Auswege an: Entweder man investiert etwas mehr für Qualitätskassetten und einen besseren Recorder mit garantiert senkrecht stehendem Magnetkopf. Oder man gibt noch mehr aus für einen digital speichernden Datenträger wie z.B. „ECMA-Kassette" oder Floppy Disk. Auch hierfür werden Aufzeichnungsverfahren vorgestellt. In einem zweiten Beitrag sind ergänzend Beispiele für Datenübertragungen, für Speicherung auf Kassette und Floppy Disk sowie für die Datensicherung angegeben.

Mit Massenspeichern kommen Benutzer entsprechend ausgestatteter Kleinrechner immer direkt in Berührung: Magnetstreifen, Kassetten oder Floppies müssen in die Transporteinrichtung geschafft und der Schreib- oder Leselauf muß gestartet werden. Das erklärt, warum Probleme und Entwicklungen in diesem Zusammenhang besonderes Interesse wecken. Halbleiterspeicher (RAM und ROM) treten selten so offen in Erscheinung. Trotzdem gibt es viele Gründe dafür, sich auch damit etwas näher zu beschäftigen. Ein Anlaß kann schon sein, daß man den preiswert auf Kassette angeschafften Assembler oder BASIC-Interpretierer in EPROM übertragen möchte.

Wie auch immer, alle Leser mit weitergehenden Interessen finden in den folgenden Aufsätzen Informationen über programmierbare Festwertspeicher, über Grundschaltungen und Herstellungstechnologien moderner Halbleiterspeicher sowie über eine wichtige Anwendung von CCD-Speichern.

Externe Massenspeicher sind für eine sinnvolle Nutzung der durch moderne Kleinrechner gebotenen Möglichkeiten unverzichtbar. Wenn man aber bedenkt, daß Einkarten-Mikrocomputer schon für 300 DM zu haben sind und beispielsweise der PET bei weniger als 2000 DM einschließlich Bildschirm und Kassettenrecorder geliefert wird, dann ist plausibel, daß ein angemessener Massenspeicher (in der Regel Kassettenrecorder) möglichst billiger als der Einkarten-$\mu$C sein muß. Im folgenden Beitrag werden ein paar relevante Eigenschaften preiswerter Massenspeicher besprochen. Aber auch „professionelle" und damit in der Regel teure Verfahren zur Speicherung digitaler Daten auf magnetischen Datenträgern werden behandelt.

H. Schumny

# Aufzeichnungsverfahren für magnetische Datenträger

Als magnetische Datenträger bezeichnet man allgemein solche Medien, die analoge oder digitale Signale so lange speichern können, bis sie durch einen ausreichend großen Löschstrom wieder in einen neutralen Zustand zurückversetzt werden. Wichtige Anwendungsfälle sind:

- Schallaufzeichnung
- Meßwertspeicherung
- Videoaufzeichnung
- Speicherung digitaler Daten

In allen vier Fällen sind heute analoge und digitale Aufzeichnungsverfahren bekannt. Schall-, Meßwert- und Videoaufzeichnungen wurden ursprünglich analog vorgenommen, werden heute aber auch digital ausgeführt **(Pulscodemodulation, PCM).** Digitale Daten dagegen wurden eigentlich immer auch digital gespeichert, mit speziellen und darum teuren Datenträgern und Laufwerken. Erst seit Einführung preiswerter Mikrocomputer und durch den Wunsch nach ebenso billigen Massenspeichern begann die Entwicklung von Aufzeichnungsverfahren für digitale Daten, die eine Verwendung gewöhnlicher Musikrecorder und Musikkassetten erlauben. Wir werden im folgenden verschiedene Aufzeichnungsverfahren besprechen und Beispiele dazu angeben. Auf die Normung wird jeweils eingegangen.

## 1 Aufzeichnungsverfahren im Tonfrequenzbereich

Aufzeichnungen im **Tonfrequenzbereich** zum Zwecke der Speicherung digitaler Daten werden aus Kostengründen nur auf Magnetbandkassetten vorgenommen. HiFi-Kassettenrecorder der gehobenen Preisklasse können unter Verwendung von z.B. Chromdioxid-Kassetten den **Hörfrequenzbereich** zwischen etwa 30 Hz und 15 kHz speichern und wiedergeben. Geräte unter 100 DM aber kommen mit Eisenoxid-Kassetten vielleicht auf einen Frequenzbereich von 300 Hz bis 5000 Hz. Aufzeichnungsverfahren müssen für diesen Fall mithin so ausgelegt sein, daß die beiden Binär-

**Pulscodemodulation:** (PCM). Verfahren zur Codierung, Speicherung und Übertragung von analogen Signalen (z.B. Musik, Sprache, Bilder), wobei die digitalisierten Signalverläufe binär codiert werden.

**Tonfrequenzbereich:** Damit wird allgemein der durch Sprache, Musikinstrumente etc. erzeugte und für das menschliche Ohr hörbare Frequenzbereich bezeichnet.

**Hörfrequenzbereich:** Wird anstelle von „Tonfrequenzbereich" benutzt, wenn beachtet werden soll, daß der hörbare Frequenzbereich mit zunehmendem Alter abnimmt.

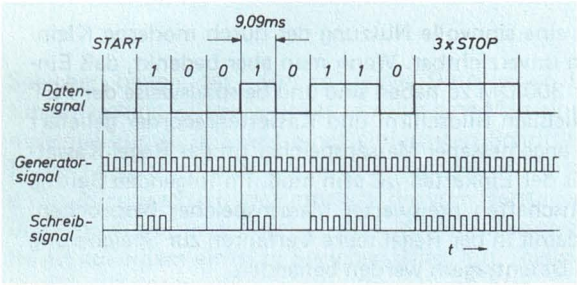



**Bild 1** TK 80-Schreibverfahren

zustände, Folgen von Null- und Einsbit also, durch Signale dargestellt werden, deren Frequenzen in diesem Bereich liegen.

Zur Darstellung der Binärzustände werden in der Regel Verfahren der **Frequenzcodierung** benutzt. Das bedeutet, den beiden Zuständen werden Schwingungen verschiedener Frequenz zugeordnet. Zu unterscheiden sind dabei zwei Untergruppen:

- *Frequenztastung* – Ein- oder Ausschaltung eines Rechteckgenerators, je nachdem, ob eine Eins oder eine Null gespeichert werden soll;
- *Frequenzumtastung* – Zuordnung einer höheren und einer niedrigeren Frequenz zu den Binärzuständen (*Frequency Shift Keying*, FSK).

Im folgenden werden ein paar Beispiele besprochen.

### TK80-Verfahren

**Bild 1** zeigt, daß es sich um eine Frequenztastung handelt. Für ein Einsbit wird ein Rechteckgenerator 9,09 ms lang eingeschaltet, für ein Nullbit ausgeschaltet. Jedes ASCII-Zeichen wird mit 8bit codiert; vorangestellt ist ein *Startbit* (logisch 0), den Abschluß bilden drei *Stopbit* (3mal logisch 1). Dieses Verfahren wird nur für relativ niedrige **Übertragungsraten** verwendet. Bei 110 Bd **(Fernschreibergeschwindigkeit)** und mit 12bit pro Zeichen ergibt sich eine Übertragung von 9 Byte oder ASCII-Zeichen pro Sekunde. Für das Schreiben oder Lesen von 1024 Byte (1 Kbyte) sind also nahezu 2 Minuten nötig.

### KIM-Verfahren

Der populäre Einkarten-$\mu$C KIM-1 verwendet – wie fast alle anderen $\mu$C – ein Frequenzumtastungsverfahren. Gemäß **Bild 2** wird jedes Bit durch eine 7,452 ms lange Aufzeichnung dargestellt, wobei im ersten Drittel immer 9 Impulse der Frequenz 3700 Hz, im letzten Drittel immer 6 Impulse der Frequenz 2400 Hz geschrieben werden. Das mittlere Drittel bestimmt den logischen Zustand: Bei 3700 Hz ist ein

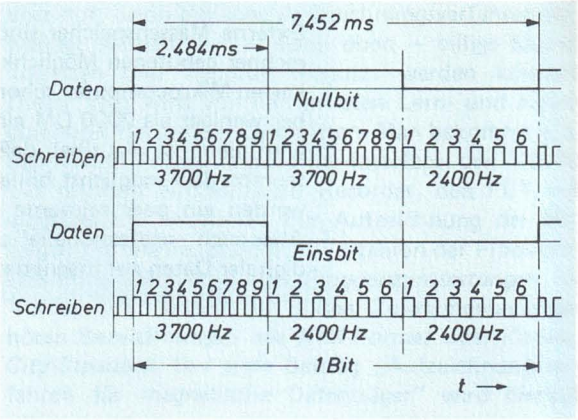



**Bild 2** KIM-Schreibverfahren

Nullbit gespeichert, bei 2400 ein Einsbit. Dieses Verfahren ist sehr sicher. 1024 Byte werden in etwa einer Minute gelesen. Durch Reduzierung der Impulszahl pro „Bitdrittel" auf minimal 2 höherfrequente und je einen niederfrequenten kann die Übertragungsrate auf das Sechsfache gesteigert werden (*Hypertape*).

### ALPHA-Verfahren

Das vom Computer ALPHA-1 benutzte Verfahren gleicht dem des KIM-1. Es wird lediglich das unterschiedliche Frequenzpaar 2400 Hz/3600 Hz verwendet.

### PET-Verfahren

Der in den PET 2001 eingebaute Kassettenrecorder benutzt das in **Bild 3** skizzierte Schreibverfahren. Danach wird mit drei Schwingungen gearbeitet:

Short: 360 $\mu$s $\hat{=}$ 2,778 kHz
Long: 510 $\mu$s $\hat{=}$ 1,960 kHz
Byte: 680 $\mu$s $\hat{=}$ 1,470 kHz

**Frequenzcodierung:** Darstellung von Signalverläufen durch Schwingungen verschiedener Frequenzen. Für analoge Signale z.B. FM (*Frequenzmodulation*), für digitale Signale z.B. FSK (*Frequency Shift Keying*).

**Übertragungsrate:** Darunter versteht man die pro Zeiteinheit übertragene Informationsmenge, z.B. 10 Zeichen pro Sekunde.

**Fernschreibergeschwindigkeit:** Spezielle Übertragungsrate der seriell sendenden Fernschreiber; angegeben in „Stromschritten pro Sekunde", abgekürzt als Bd (*Baud*) bezeichnet. Weil jedes Fernschreibzeichen aus 11 Stromschritten (11 bit) besteht, entspricht der Fernschreibgeschwindigkeit von 110 Bd eine Übertragungsrate von 110 bit/s bzw. 10 Zeichen/s.

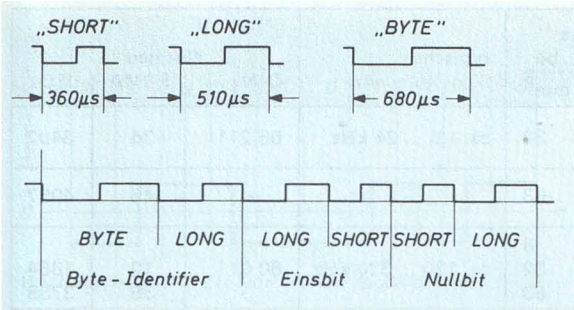



**Bild 3** PET-Schreibverfahren

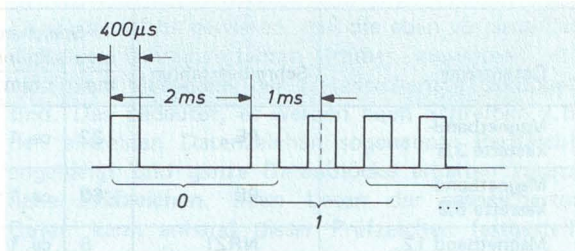



**Bild 4** TRS 80-Schreibverfahren

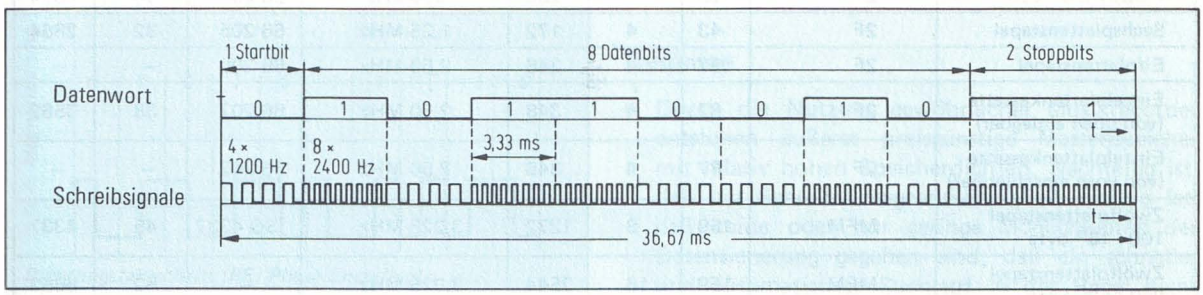



**Bild 5** Kansas-City-Standard

Die Folge "Short + Long" bedeutet „Nullbit", die Folge "Long + Short" bedeutet „Einsbit". Jedes Datenbyte wird aus solchen Folgen zusammengesetzt. Vorangestellt ist als "Byte Identifier" eine Folge "Byte + Long". Damit ergibt sich für ein ganzes Byte eine „Zeichenlänge" von nur 8,15 ms. Das ist weniger als bei der sechsfachen KIM-Schreibdichte, die **Bit-Packungsdichte** ist hier also noch größer.

### TRS 80-Verfahren

Der Hobby-Computer TRS 80 von Tandy, Radio Shack, arbeitet mit einem Verfahren, daß schon sehr den digitalen Speichermethoden ähnelt. **Bild 4** macht deutlich, daß eine hohe Bit-Packungsdichte erzielt wird, die der etwa vierfachen KIM-Schreibdichte entspricht.

### Kansas-City-Standard

Mit den eben beispielhaft besprochenen Aufzeichnungsverfahren ist deutlich geworden, daß praktisch jeder Hersteller eine andere Methode benutzt, ein Austausch von Datenträgern (Kassetten) zwischen verschiedenen µC-Systemen damit praktisch unterbunden ist. 1975 wurde darum in Kansas City, USA, von amerikanischen µC-Herstellern ein Aufzeichnungsverfahren vorgeschlagen, daß nun als *KC-Standard* bekannt ist. Nach **Bild 5** ist jede Bitzelle dabei 3,33 ms lang. Einsbit werden durch acht 2400 Hz-Impulse dargestellt, Nullbit durch vier 1200 Hz-Impulse.

Die nachfolgende Tabelle ist eine Zusammenstellung der bei den eben besprochenen Schreibverfahren erzielten **Speicherdichten,** wenn die für Musikrecorder genormte Bandgeschwindigkeit von 4,75 cm/s einheitlich angenommen wird.

| Verfahren | Bitlänge in ms | Bytelänge in ms | Speicherdichte bit/mm | Speicherdichte Zeichen/cm |
|---|---|---|---|---|
| TK 80 | 9,09 | 109,08 | 2,3 | 1,9 |
| KIM (Normal) | 7,452 | 81,97 | 2,8 | 2,5 |
| KIM (Super) | 2,484 | 37,32 | 8,5 | 7,7 |
| KIM (Hyper) | 1,242 | 13,66 | 17 | 15,4 |
| PET | 0,87 | 8,15 | 24 | 25,8 |
| TRS-80 | ca. 2 | ca. 20 | 10,5 | 10,5 |
| KC | 3,33 | 36,67 | 6,3 | 5,7 |

**Bit-Packungsdichte:** Anzahl der in einer Aufzeichnungsspur gespeicherten (gepackten) Bit pro Längeneinheit.

**Speicherdichte:** Anzahl der gespeicherten Informationsmenge pro Längeneinheit, also z.B. 2,3 bit/mm entsprechend 1,9 Zeichen/cm, weil in diesem Beispiel jedes Zeichen aus 12 bit besteht.

| Datenträger | Schreibverfahren | Speicherdichte bit/mm | Spuren/mm | bit/mm² | typische Signalfrequenz | Normen DIN | ECMA | ISO |
|---|---|---|---|---|---|---|---|---|
| Magnetbandkassette 3,8 | PE | 32 | ca. 1 | 32 | ca. 1,5 ... 24 kHz | 66 211 | 34 | 3407 |
| Magnetbandkassette 6,3 | PE | 63 | ca. 1 | 63 | | – | 46 | 4057 |
| Magnetband 12 | NRZI<br>NRZI<br>PE | 8<br>32<br>63 | ca. 1<br>1<br>1 | 8<br>32<br>63 | ca. 120 ... 320 kHz | 60 011 | 12<br>36 | 1864<br>3788 |
| Magnetband 12 mit GCR | | 246 | ca. 1 | 246 | z.B. 720 kHz | ISO 5652 | | 5652 |
| Floppy Disk 200 | 2F | 128 | 2 | 256 | 250 kHz | 66 237 | 54 | 5654 |
| Sechsplattenstapel | 2F | 43 | 4 | 172 | 1,25 MHz | 66 205 | 32 | 2864 |
| Elfplattenstapel | 2F | 87 | 4 | 348 | 2,50 MHz | 66 206 | – | – |
| Einzelplattenkassette (von oben einlegbar) | 2F | 87 | 4 | 348 | 2,50 MHz | 66 207 | 38 | 3562 |
| Einzelplattenkassette (von vorn einschiebbar) | 2F | 87 | 4 | 348 | 2,50 MHz | 66 242 | – | – |
| Zwölfplattenstapel $100 \cdot 10^6$ byte | MFM | 159 | 8 | 1272 | 3,225 MHz | ISO 4337 | 45 | 4337 |
| Zwölfplattenstapel $200 \cdot 10^6$ byte | MFM | 159 | 16 | 2544 | 3,225 MHz | – | 52 | 5653 |
| Neue Magnetplatten | | 248 | 19 | 4712 | 4,792 MHz | – | – | – |

## 2 Digitale Aufzeichnungsverfahren

Mit digitalen Aufzeichnungsverfahren meint man die nun schon „klassischen" Methoden der Speicherung digitaler Daten auf Computerbändern, Kassetten, harten Magnetplatten und Floppy Disks. Genutzt werden die nach DIN 66 010 genormten Schreibverfahren (*Recording Modes*), die in **Bild 6** auszugsweise wiedergegeben sind.

Die obenstehende Tabelle ist eine Zusammenstellung der wichtigsten genormten Datenträger. Angegeben sind Speicherdichten in bit/mm und in bit/mm². Diese **Flächenspeicherdichte** macht natürlich nur einen Sinn, wenn in mehreren nebeneinander liegenden Spuren aufgezeichnet wird. Ebenfalls aufgeführt sind typische Signalfrequenzen beim Schreiben und Lesen der Daten, die nach Norm empfohlenen Schreibverfahren und die relevanten Normen. Die Abkürzungen bedeuten:

3,8: Kassette mit 3,81 mm breitem Magnetband (sogenannte *Philips-Kassette*)

6,3: Kassette mit 6,30 mm breitem Magnetband (sogenannte *3M-Kassette*)

12: *Computerband* der Breite 12,7 mm

GCR: *Group Coded Recording* (neues Speicherverfahren mit Gruppencodierung)

DIN: *Deutsches Institut für Normung*

ECMA: *European Computer Manufacturers Association*

ISO: *International Organization for Standardization*

MFM: *Modified Frequency Modulation*

Die in der Tabelle aufgeführte Floppy Disk (Flexible Magnetplatte) mit 200 mm Durchmesser ist als Standardtyp anzusehen. Sie wird nur einseitig genutzt und verfügt dabei über eine „Brutto"-Speicherkapazität (**unformatiert** also) von 250 Kbyte. Inzwischen werden auch beide Seiten benutzt und mit doppelter Dichte beschrieben, so daß die erstaunliche Datenmenge von bis zu ca. $10^6$ byte (1 Mbyte) abgelegt werden kann.

**Flächenspeicherdichte:** Ergibt sich aus der Anzahl der Bit pro Millimeter mal der Zahl der in einem Millimeter nebeneinander angeordneten Aufzeichnungsspuren.

**Formatierung:** Vom Hersteller oder durch Normen festgelegtes Verfahren zur Anordnung von Daten, Datenblöcken, Steuerzeichen etc. auf einem Datenträger. Durch Formatierung geht (je nach Format) die Brutto-Kapazität auf bis zu 20 % zurück.

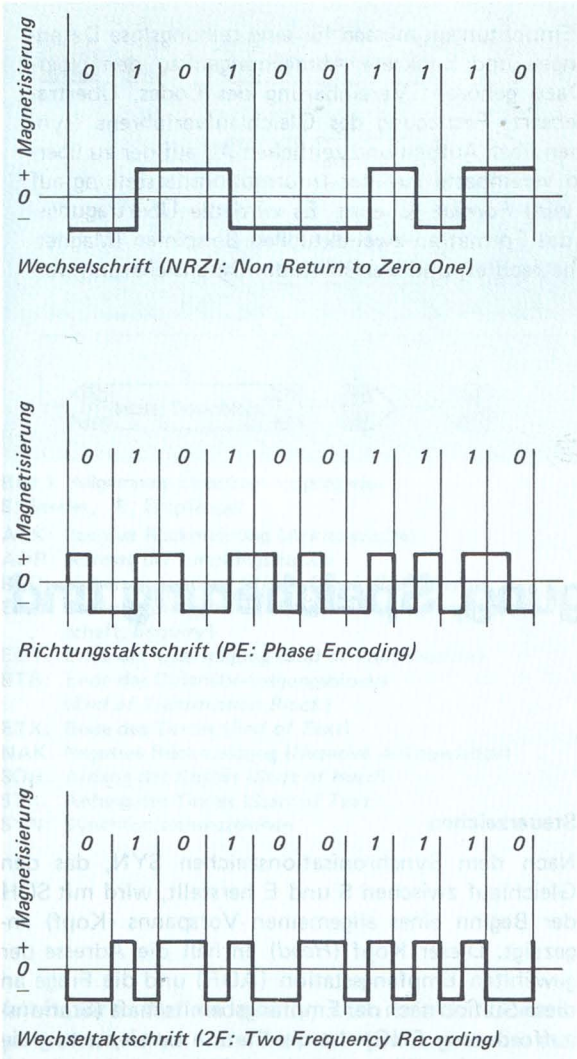



**Bild 6** Digitale Schreibverfahren nach DIN 66 010

Für µC-Anwendungen von größerer Bedeutung sind die sogenannten "Mini Floppies". Das sind praktisch auf einen Außendurchmesser von 130 mm verkleinerte Standard-Floppies mit Brutto-Speicherkapazitäten zwischen 80 und 320 kbyte. Für die Textverarbeitung wird inzwischen eine "Euro-Diskette" genannte Version mit 105 mm Durchmesser angeboten. Ferner existiert eine als Lochstreifenersatz an Fernschreibern deklarierte kleine Flexible Magnetplatte mit 65 mm Durchmesser.

Es sei darauf hingewiesen, daß die eben vorgestellten digitalen Schreibverfahren immer zusammen mit speziellen Methoden der *Datensicherung* gekoppelt sind. Das bedeutet, es werden beim Schreiben z.B. den einzelnen Datenzeichen sogenannte **Paritätsbit** angehängt und ganze Datenblöcke erhalten zusätzliche **Prüfzeichen**. Beim Lesen der gespeicherten Daten kann anhand dieser Prüfzeichen festgestellt werden, ob Übertragungs- bzw. Schreib-Lesefehler aufgetreten sind. Die Benutzung spezieller Codes ermöglicht gar eine automatische Korrektur erkannter Fehler **(ECC: Error Correcting Codes)**.

## Resümee

- Durch die Nutzung gewöhnlicher Musikrecorder entstehen äußerst preisgünstige Massenspeicher mit relativ hohen Speicherdichten. Nachteilig ist, daß die Schreib-Lesegeschwindigkeit niedrig ist; daß keine oder nur geringe Möglichkeiten der Datensicherung gegeben sind; daß ein schneller und automatischer Suchlauf in der Regel nicht existiert.
- Massenspeicher mit digitalen Datenträgern (Kassetten oder Floppies) sind erheblich teurer und erfordern besondere Maßnahmen (**Controller, Formatter** etc.). Die Vorteile für den Benutzer sind aber unübersehbar: Hohe Schreib-Lesegeschwindigkeiten, sehr große Speicherdichten, schneller Suchlauf, sehr zuverlässige Datensicherungen, insgesamt also ein großer Komfort verbunden mit hoher Sicherheit.

**Paritätsbit:** Für Paritätsprüfungen wird jedes Zeichen mit Hilfe eines zusätzlichen Paritätsbit auf eine (je nach Vereinbarung) gerade oder ungerade Quersumme ergänzt.

**Prüfzeichen:** Ergeben sich, ähnlich wie Paritätsbit, nach verschiedenen Verfahren der Quersummenbildung (*Längsparität, Querparität* etc.).

**ECC:** *Error Correcting Code*. Vorschrift zur Bildung spezieller Prüfzeichen, mit deren Hilfe eine automatische Fehlerkorrektur in der Lesestation möglich wird.

**Controller:** Hier spezielle elektronische Einheit, die das Schreiben auf und Lesen von Kassetten oder Floppies steuert.

**Formatter:** Elektronische Einheit (oft mit dem Controller verknüpft) zur richtigen Formatierung oder auch zur Formatumsetzung.

Neben den nötigen technischen Einrichtungen müssen für eine reibungslose Datenübertragung Sicherungseinrichtungen und konkrete Abmachungen zu den Übertragungsprozeduren bestehen. Dazu gehören: Vereinbarung des Codes, Übertragungsgeschwindigkeit und Betriebsart, Festlegung des Gleichlaufverfahrens (synchron, asynchron) und Absprachen über Aufbau und zeitlichen Ablauf der zu übertragenden Informationen. Die so vereinbarte Art der Informationsdarstellung auf den verwendeten Datenträgern wird *Format* genannt. Es wird die Übertragungsprozedur allgemein besprochen, das Format an zwei aktuellen Beispielen (Magnetbandkassette und Floppy Disk) betrachtet, und Verfahren der Datensicherung werden diskutiert.

H. Schumny

# Beispiele für die Übertragung, Speicherung und Sicherung von Daten *)

## 1 Übertragungsprozedur

Alle Regeln, nach denen der Verkehr zwischen zwei Datenstationen ablaufen muß, werden mit der *Übertragungsprozedur* festgelegt, die neben den Informationen (Daten) selbst eine Reihe von Steuerzeichen und Gleichlaufanweisungen enthalten muß. Die zu übertragenden Informationen können aus einzelnen Zeichen bestehen, einen *Datensatz* oder einen *Datenblock* als Summe von Datensätzen bilden.

### Beispiel 1: Synchrone Stapelübertragung

Mit **Bild 1** ist der einfache Fall einer Übertragungsprozedur für das Beispiel „Synchrone Stapelübertragung in mehreren Datenblöcken" angegeben. In dieser stark vereinfachten Darstellung sind die zeitlich nacheinander ablaufenden Sender- und Empfängerinformationen S und E durch Richtungssymbole gekennzeichnet.

### Steuerzeichen

Nach dem Synchronisationszeichen SYN, das den Gleichlauf zwischen S und E herstellt, wird mit SOH der Beginn eines allgemeinen Vorspanns (Kopf) angezeigt. Dieser Kopf (*Head*) enthält die Adresse der gewählten Empfangsstation (ADR) und die Frage an diese Station nach der Empfangsbereitschaft (Stationsaufforderung ENQ). Ist die Station bereit, sendet sie eine positive Rückmeldung (ACK), wobei auch das Synchronisationszeichen vorangestellt ist, wie überhaupt zu Beginn jeder Übertragung dieses Zeichen stehen muß. Auf die positive Antwort hin kann die Sendestation mit der Datenübertragung beginnen, wobei wieder das Synchronisationszeichen und das Steuerzeichen STX (Anfang des Textes) vorangestellt sind. Das Ende der Übertragung des ersten Datenblocks wird mit ETB (Ende des Datenübertragungsblocks) angezeigt.

### Sicherungszeichen

Das oder die Sicherungszeichen (BCC, vgl. Abschnitt 4) bilden den endgültigen Abschluß eines Datenblocks. Hierauf wird eine Rückmeldung des Empfängers erwartet, die positiv (ACK) oder negativ (NAK) aus-

*) Auszug aus *H. Schumny,* Signalübertragung, Verlag Vieweg, Braunschweig/Wiesbaden

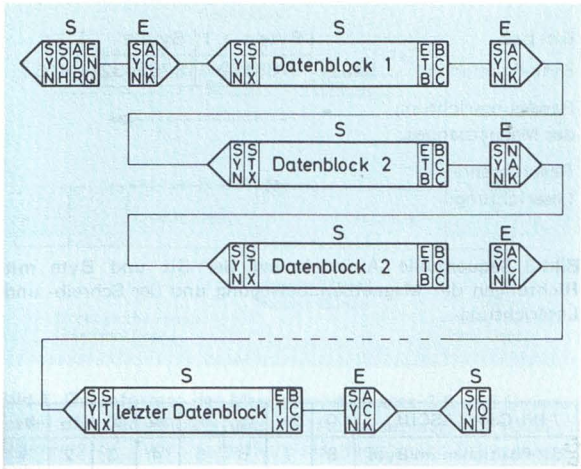



**Bild 1** Allgemeine Übertragungsprozedur
S: Sender, E: Empfänger
ACK: Positive Rückmeldung (*Acknowledge*)
ADR: Adresse der Empfangsstation
BCC: Sicherungszeichen (*Block Check Character*)
ENQ: Stationsaufforderung (Frage nach Empfangsbereitschaft, *Enquiry*)
EOT: Ende der Übertragung (*End of Transmission*)
ETB: Ende des Datenübertragungsblocks (*End of Transmission Block*)
ETX: Ende des Textes (*End of Text*)
NAK: Negative Rückmeldung (*Negative Acknowledge*)
SOH: Anfang des Kopfes (*Start of Head*)
STX: Anfang des Textes (*Start of Text*)
SYN: Synchronisationszeichen

fallen wird. War sie positiv, kann mit der Übertragung des zweiten Datenblocks begonnen werden etc. Für die negative Rückmeldung sind zwei Ursachen möglich: Einmal kann sie anzeigen, daß der Empfänger nicht in Bereitschaft ist. Dann muß die Sendestation so lange weiter anfragen, bis die Rückmeldung positiv wird. Zweitens kann sie aufgrund der Analyse des Sicherungszeichens BCC den fehlerhaften Empfang der Information melden (Fehlerquittung NAK). Daraufhin muß der fehlerhaft empfangene Block erneut übertragen werden usw., wie das für den Datenblock 2 in **Bild 1** gezeigt ist.

### Ende der Übertragung

Der letzte übertragene Datenblock wird mit dem Steuerzeichen ETX (Ende des Textes) abgeschlossen. Den endgültigen Abbruch der Übertragung zeigt das Zeichen EOT (Ende der Übertragung) an.

### Beispiel 2: HDLC-Prozedur

In den letzten Jahren ist international eine leistungsfähige Datenübertragungsprozedur genormt worden (ISO 3309.2 und 4335, ECMA-40 und 49, DIN 66 221). Es handelt sich um ein *bitorientiertes Steuerungsverfahren bei synchroner Datenübertragung*, das die internationale Kurzbezeichnung *HDLC (High Level Data Link Control)* trägt. Das Verfahren zeichnet sich gegenüber der in Beispiel 1 vorgestellten Basis-Prozedur vor allem aus durch:

- bessere Leitungsausnutzung (Vollduplex-Betrieb);
- optimale Datensicherung durch Verwendung selbstkorrigierender Codes (CRC bzw. ECC, vgl. Abschnitt 4);
- geringere Anzahl von Steuerzeichen;
- weitgehende Unabhängigkeit von der Netzform und der „Intelligenz" der Terminals.

Der Aufbau eines Datenübertragungsblocks (*Frame Structure*) ist wie folgt:

| Block-begrenzung *(Flag)* | Adreßfeld *(Address)* | Steuerfeld *(Control)* | Datenfeld *(Information)* | Block-prüfungsfeld *(FCS)* | Block-begrenzung *(Flag)* |
|---|---|---|---|---|---|
| 01111110 | 8 bit | 8 bit | *n* bit | 16 bit | 01111110 |

Das *Datenfeld* enthält eine beliebige Anzahl von Datenbit, wobei aber *n* in der Regel ein Vielfaches von 8 ist. Sollen lediglich Befehle oder Meldungen übertragen werden, wird $n = 0$; das Datenfeld entfällt also (*Frames containing Data Link Supervisory Information only*). DÜ-Blöcke beginnen und enden stets mit der *Blockbegrenzung* (01111110), die auch zur Synchronisierung der Datenendeinrichtungen dient. Das *Adreßfeld* enthält immer die Identifizierung der Datenstation. Das *Steuerfeld* enthält Befehle oder Meldungen und Blocknummern und dient der Leitsteuerung. Die *Blockprüfzeichenfolge* (*Frame Checking Sequence*, FCS) besteht aus 16 bit. Sie wird nach einem *zyklischen Verfahren* berechnet, unter Einbeziehung sämtlicher Bit des DÜ-Blocks, außer den 2 X 8 bit der Blockbegrenzungen.

## 2 Format: Magnetbandkassette

Für die Übertragung von Daten werden festgelegte Prozeduren benötigt, in denen die Daten selbst von Steuerzeichen umrahmt sind **(Bild 1)**, die vom Sende- und Empfangsgerät erzeugt werden. Bei der Übertragung von und auf maschinell lesbare Datenträger müssen auch auf diesen die Daten in feststehenden

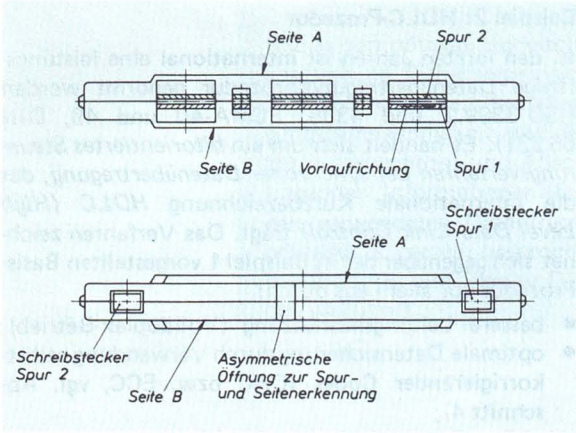



**Bild 2** Magnetbandkassette 3,8 mit Spurlage (Spur 1 und 2) und Seitenerkennungsschlitz (Asymmetrische Öffnung) für Kassettenseiten A und B; Darstellung nach DIN 66 212

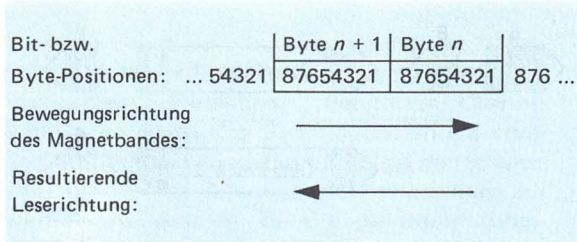



**Bild 3** Sequentielle Aufzeichnung der Bit und Byte mit Richtungen der Magnetbandbewegung und der Schreib- und Leserichtung

| 7-Bit-Code (ASCII) | 0 | $b_7$ | $b_6$ | $b_5$ | $b_4$ | $b_3$ | $b_2$ | $b_1$ |
|---|---|---|---|---|---|---|---|---|
| Bit-Positionen im Byte | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 |

| 8-Bit-Code (EBCDIC) | $b_8$ | $b_7$ | $b_6$ | $b_5$ | $b_4$ | $b_3$ | $b_2$ | $b_1$ |
|---|---|---|---|---|---|---|---|---|
| Bit-Positionen im Byte | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 |

**Bild 4** Zuordnung der Datenbit zu den Codebit

Anordnungen gespeichert sein. D.h. die einzelnen Datenblöcke müssen in einem vereinbarten *Datenformat* auf dem jeweiligen Träger angeordnet sein – kurz *Format* genannt. Das Anordnen der Daten in festgelegter Folge und Art heißt *Formatierung.*

### Normung

Datenformate sind für alle Medien für Datenerfassung und -eingabe vereinbart und international in Normen festgelegt. Hier sei zunächst am Beispiel der Magnetbandkassette 3,8, die wegen ihrer Kompaktheit besonders für den Datenaustausch geeignet ist, solch ein Format vorgestellt. Die zugehörigen Normblätter sind in Deutschland DIN 66 211 und DIN 66 212, international ISO 3407 sowie ECMA-34 und ECMA-41. (ECMA: *European Computer Manufacturers Association*.)

### Datendarstellung

Das Abspeichern von Daten kann, wie in **Bild 2** angedeutet, in zwei Spuren 1 und 2 geschehen (Seite A bzw. B oben). Für den Datenaustausch ist Spur 1 reserviert; Spur 2 kann in diesem Fall nur nach besonderer Vereinbarung benutzt werden, worauf dann auch eigens hingewiesen werden muß. Die Datendarstellung erfolgt ganz allgemein bitseriell, aber in Einheiten zu jeweils einem Byte geordnet. **Bild 3** zeigt, wie alle Informationen bit- und byteweise gespeichert und gelesen werden, wobei das Bit mit der niedrigsten Wertigkeit (also Bit 1) immer zuerst geschrieben und gelesen wird.

### Codierung

Vereinbart sind der 7-Bit-Code (ASCII) und ein 8-Bit-Code (z.B. EBCDIC). **Bild 4** gibt die Zuordnungen der einzelnen Datenbit zu den Codebit an.

### Datenblock

Mehrere Zeichen werden zu einem Datenblock zusammengefaßt, der aus *Präambel*, Datenteil, CRC-Prüfzeichen und *Postambel* zusammengesetzt ist **(Bild 5)**. Prä- und Postambel bestehen jeweils aus der Bitkombination „10101010", wobei das Schreiben und Lesen von rechts, also mit „0" beginnt. Diese Zeichen dienen zum Kennzeichnen von Beginn und Ende eines Datenblocks. Der Datenteil selbst kann minimal 16, maximal 2048 Bit betragen. Das CRC-Prüfzeichen ist 16 Bit lang; es wird in Abschnitt 4 näher erläutert.

### Zwischenräume

Der Datenblock ist eine zusammenhängende, selbständige Einheit. Die Anordnung innerhalb der Aufzeichnungsspur ist ebenfalls in **Bild 5** angegeben. Am Beginn des Magnetbandes muß zwischen dem Bandbeginn-Erkennungsloch (BOT = *Beginning of Tape*) und dem ersten Datenblock ein Leerraum von

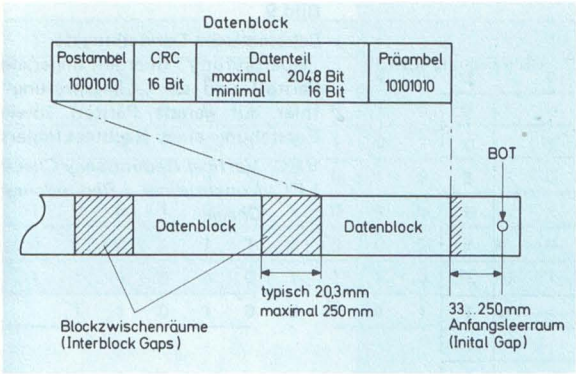



**Bild 5** Datenformat der Magnetbandkassette 3,8 und Organisation der Datenblöcke und der Aufzeichnungsspur (BOT = *Beginning of Tape*, ausgeführt als Bandbeginn-Erkennungsloch)

33 ... 250 mm gelassen werden (*Initial Gap*). Zwischen zwei Datenblöcken muß ein Zwischenraum (*Interblock Gap*) von typisch 20,3 mm, aber maximal 250 mm frei bleiben. Jeder Zwischenraum von mehr als 400 mm wird vom Lesegerät als Ende der Datenaufzeichnung in dieser Spur interpretiert. Damit ist ein einfaches Beispiel für eine Formatierung gegeben. Mit Hilfe solch eines Datenformats werden die in **Bild 1** gezeichneten Datenblöcke ausgefüllt. Ein erheblich komplizierteres Format wird im nächsten Abschnitt besprochen.

## 3 Format: Flexible Magnetplatte

Anders als beim sequentiellen Datenträger Magnetbandkassette werden auf der *Floppy Disk* (Flexible Magnetplatte) Daten in konzentrischen Spuren gespeichert **(Bild 6a)**, so daß ein quasidirekter Zugriff

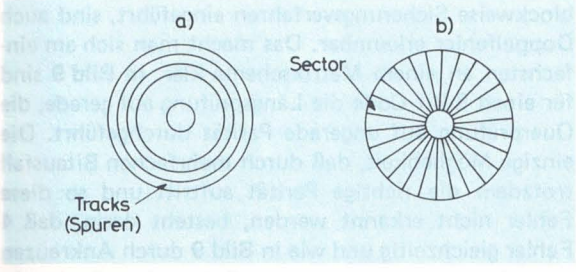



**Bild 6** Spuren und Sektoren auf der Flexiblen Magnetplatte (*Flexible Disk*)

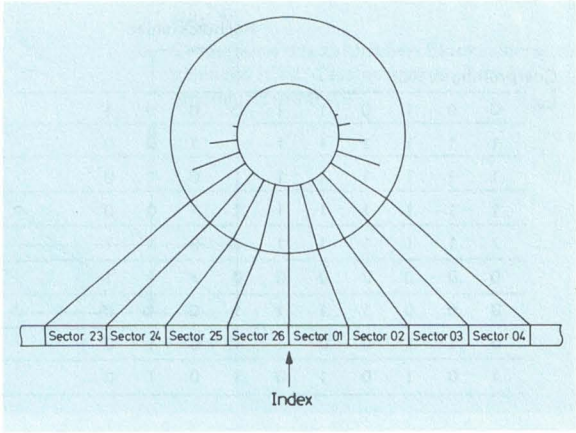



**Bild 7** Numerierung der 26 Sektoren 01 bis 26 über alle Spuren 00 bis 76 hin

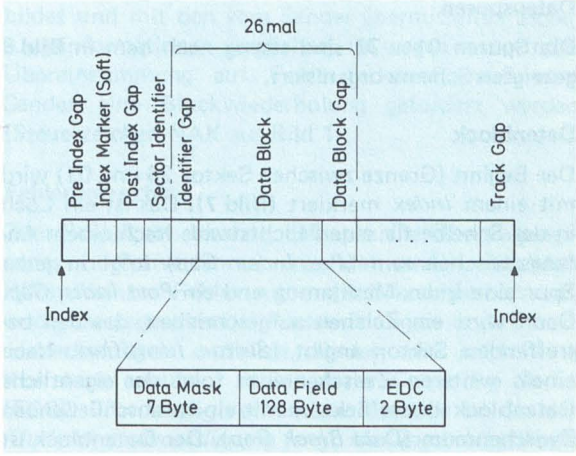



**Bild 8** Datenformat einer Flexiblen Magnetplatte und Organisation der Datenblöcke

zu den Daten entsteht. Das Abspeichern ist in insgesamt 77 Spuren möglich, die gemäß **Bild 6b** in 26 *Sektoren* eingeteilt sind. Die Numerierung der Spuren erfolgt von 00 ... 76, die der Sektoren von 01 ... 26 **(Bild 7)**.

### Index-Spur

Die Spur 00 (*Directory* oder *Index Track*) kann nicht mit Daten belegt werden. Sie dient zur Kennzeichnung, enthält sozusagen ein Inhaltsverzeichnis. Außerdem ist es hierin möglich, bis zu zwei eventuell zerstörte Spuren zu identifizieren und Ersatzspuren zuzuordnen, so daß nur maximal 74 Datenspuren zur Verfügung stehen.

Rechteckfehler

Querprüfung (VRC) Längsprüfung (LRC)

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 1 |
| 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |

**Bild 9**
Schematische Darstellung der „Querprüfung" (hier auf ungerade Parität) und der „Längsprüfung" (hier auf gerade Parität) sowie Entstehung eines Rechteckfehlers
VRC: *Vertical Redundancy Check*
LRC: *Longitudinal Redundancy Check*

### Datenspuren

Die Spuren 01 ... 76 sind streng nach dem in **Bild 8** gezeigten Schema organisiert.

### Datenblock

Der Beginn (Grenze zwischen Sektor 26 und 01) wird mit einem *Index* markiert **(Bild 7).** Das ist ein Loch in der Scheibe für einen Lichtstrahl. Nach einem Anfangszwischenraum (*Pre Index Gap*) folgt in jeder Spur eine Index-Markierung und ein *Post Index Gap.* Dann wird ein Zeichen aufgeschrieben, das den betreffenden Sektor angibt (*Sector Identifier*). Nach einem weiteren Zwischenraum folgt der eigentliche Datenblock dieses Sektors mit einem abschließenden Zwischenraum (*Data Block Gap*). Der Datenblock ist seinerseits unterteilt in ein *Data Mask* (7 Byte), das Datenfeld (128 Byte) und ein EDC-Prüfzeichen (2 Byte).

## 4 Datensicherung

Von größter Bedeutung für eine reibungslose Datenübertragung und -speicherung sind Methoden der Datensicherung, Fehlererkennung und Fehlerkorrektur. Im einfachsten Fall muß die Erkennung und Korrektur von Fehlern bereits beim Erfassen und Eintasten von Daten in die Übertragungsstation durch den Bediener geschehen. Ebenso sind Sichtkontrollen beim Empfang üblich. Für eine definierte Sicherheit der Übertragung sorgt die Post, die für die verschiedenen Übertragungswege mittlere Schrittfehlerwahrscheinlichkeiten angibt.

### Mehrfachübertragung

Zur Erzielung einer sicheren Übertragung kann eine Information mehrfach (mindestens dreimal) gesendet werden. Die Information, die am häufigsten gleichlautend ankommt, gilt dann als fehlerfrei. Dieses Verfahren ist jedoch sehr zeitaufwendig. Ist aber eine hinreichende *Redundanz* vorhanden, können spezielle Prüfverfahren eingeführt werden.

### Paritätsprüfung

Dabei handelt es sich darum, daß jedem codierten Zeichen ein Prüfbit angehängt wird. Ist *gerade Parität* vereinbart, muß die „Quersumme" sämtlicher Bit einschließlich des Prüfbit eine gerade Zahl ergeben. Analoges gilt für *ungerade Parität*. Mit diesem Verfahren kann jeweils ein Bitausfall oder ein durch eine Störung zusätzlich entstandenes Bit pro Zeichen erkannt werden. Das gleichzeitige Auftreten zweier Fehler ist so aber nicht feststellbar.

### Rechteckfehler

Werden neben der zeichenweisen Sicherung zusätzlich blockweise Sicherungsverfahren eingeführt, sind auch Doppelfehler erkennbar. Das macht man sich am einfachsten an einem Matrixschema klar. In **Bild 9** sind für einen 8-Bit-Code die Längsprüfung auf gerade, die Querprüfung auf ungerade Parität durchgeführt. Die einzige Möglichkeit, daß durch mehrfachen Bitausfall trotzdem die richtige Parität auftritt und so diese Fehler nicht erkannt werden, besteht darin, daß 4 Fehler gleichzeitig und wie in **Bild 9** durch Ankreuzen kenntlich gemacht auftreten. Solch ein „Rechteckfehler" ist aber ein sehr unwahrscheinliches Ereignis.

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 |
| 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |
| 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 |
| 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 |
| 1 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 |
| 0 | 1 | 0 | 0 | 1 | 1 | | | | | | | |

**Bild 10**
Entstehung des zyklischen Blocksicherungszeichens (CRC-Zeichen) durch Paritätsergänzung unter 45°

### CRC-Prüfzeichen

Will man trotzdem noch weitergehende Sicherheiten vorsehen, stehen „zyklische Blocksicherungen" zur Verfügung (CRC, *Cyclic Redundancy Check*). Dieses komplizierte Verfahren kann hier nur vereinfacht vorgestellt werden. Aus **Bild 10** erkennt man, daß die gleiche Paritätsprüfung, wie man sie quer und längs vorgenommen hat, auch unter 45° durchgeführt werden kann. Im gezeigten Beispiel ist auf gerade Parität ergänzt. Man erhält so ein Binärzeichen, dessen Länge aus Blocklänge minus 1 plus Spurenzahl entsteht, im verwendeten Beispiel also aus 13 − 1 + 8 = 20. Dieses (bei langen Blöcken) vielstellige Binärzeichen wird nach vereinbarten Rechenregeln auf z.B. 9 Stellen (beim 9-Spur-Magnetband) oder 16 Stellen (bei Magnetbandkassette und Floppy Disk) umgerechnet. Das so entstandene CRC-Prüfzeichen wird am Ende des Datenblocks abgespeichert.

### Fehlererkennung

Beim Übertragen von Daten werden die Paritätsbit der Querprüfung (VRC, *Vertical Redundancy Check*) sowie die Prüfzeichen aus der zyklischen Blockprüfung (CRCC, *Cyclic Redundancy Check Character*) und der Längsprüfung (LRCC, *Longitudinal Redundancy Check Character*) mit ausgesendet. In der Empfangsstation werden die Prüfzeichen nachgebildet und mit den vom Sender übermittelten Sicherungsinformationen verglichen. Ist wegen mangelnder Übereinstimmung auf Fehler erkannt, muß vom Sender eine Blockwiederholung gefordert werden (Steuerzeichen NAK aus **Bild 1**).

### Fehlerkorrektur

Zum Schluß sei erwähnt, daß spezielle Übertragungscodes und Prüfverfahren verwendet werden, mit deren Hilfe erkannte Fehler automatisch korrigierbar sind, wodurch zeitraubende Rückmeldungen und Blockwiederholungen entfallen. Codes, die solch eine Korrektur erlauben, heißen *Error Correcting Codes* (ECC). Eine häufig verwendete Methode ist die der *Ähnlichkeitsdecordierung* (engl. *Maximum Likelihood Detection*). Dabei wird davon ausgegangen, daß ein empfangenes Zeichen, das nicht im verwendeten Code vorhanden ist, infolge einer Störung wahrscheinlich aus einem Zeichen entstanden ist, von dem es sich am wenigstens unterscheidet. Solche und viele weit raffiniertere Verfahren ermöglichen heute eine weitgehend fehlerfreie Datenübertragung.

Aktuelle Marktübersichten belegen, daß Kernspeicher immer noch eine Rolle spielen, die allerdings ständig geringer wird. Diese Übersichten betreffen dann aber den gesamten Bereich der digitalen Datenverarbeitung. Taschenrechner und Mikrocomputer sind jedoch von Anfang an ausschließlich mit Halbleiterspeichern ausgerüstet. Es muß dabei unterschieden werden zwischen Schreib-Lesespeichern (RAM) und Festwerspeichern (ROM). Für beide Typen und deren Modifikationen gilt die erstaunliche Tatsache, daß sie immer preiswerter werden und gleichzeitig die Speicherkapazität zunimmt. Der folgende Beitrag gibt einen Überblick und erklärt an Beispielen die wichtigsten Prinzipien.

O. Neufang

# Halbleiterspeicher

Durch die Entwicklung neuer Technologien wurde es in den letzten Jahren möglich, hochintegrierte Halbleiterbausteine zu entwickeln. Ein Typ dieser Bausteine sind die Halbleiterspeicher, die zur Speicherung digitaler Informationen dienen. Pro Bit sind sie billiger und in der Zugriffs- bzw. Zykluszeit schneller als die Kernspeicher. Die Umsatzzahlen für Halbleiterspeicher werden die Umsatzzahlen von Kernspeichern in naher Zukunft übertreffen.

Man kann Halbleiterspeicher in die drei Hauptkategorien RAM (*Random Access Memory,* Schreib-Lesespeicher), ROM (*Read Only Memory,* Festwertspeicher) und serielle Speicher unterteilen (s. **Bild 1**). Mit Ausnahme von CCD (*Charge Coupled Devices,* ladungsgespeicherte Einheiten) und EPROM (*Erasable Programmable Read Only Memory*), die nur in der unipolaren MOS-Technologie (MOS: *Metal Oxide Semiconductor*) gefertigt werden können, konkurrieren Bipolar- und Unipolartechniken. Bei bipolaren Techniken tragen zum elektrischen Stromfluß zwei Ladungsträgertypen (Elektronen *und* Defektelektronen) bei, während bei den unipolaren Techniken nur ein Ladungsträgertyp (Elektronen *oder* Defektelektronen) zum elektrischen Stromfluß beitragen. In den letzten Jahren wurde wegen der höheren Packungsdichte den unipolaren gegenüber den bipolaren Techniken der Vorzug gegeben. In jüngster Zeit hat es den Anschein, als könnte die im Jahre 1972 entwickelte Bipolartechnik $I^2L$ (*Integrated Injection Logic*) eine Alternative zu den Unipolartechniken sein. Ähnlich hohe Packungsdichten und kürzere Zugriffszeiten versprechen in

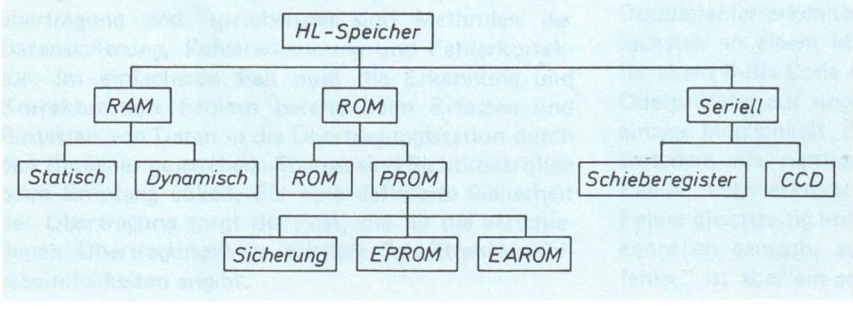



**Bild 1**
Klassifizierung von Halbleiter (HL)-Speichern

den nächsten Jahren einen interessanten Wettlauf zwischen beiden Techniken.

## 1 Schreib-Lese-Speicher (RAM)

Bei einem Schreib-Lese-Speicher kann eine binäre Information in jedes beliebige Bit eingeschrieben oder aus einem Bit ausgelesen werden. Schreib-Lese-Speicher werden in statische und dynamische RAM unterteilt. Statische RAM können eine einmal gespeicherte Information so lange festhalten, bis sie durch äußere Einwirkung veranlaßt werden, ihren Speicherinhalt zu ändern. Sie haben allerdings den Nachteil, daß bei Ausfall der Spannungsversorgung die Informationen verloren sind. Man nennt sie deshalb auch *flüchtige Speicher.* Es sind heute schon *nichtflüchtige Halbleiterspeicher* entwickelt. Da die Erfahrung mit diesen Bausteinen aber noch nicht ausreichend fundiert ist, werden sie an dieser Stelle nicht besprochen. Dynamische RAM können einmal eingespeicherte Daten auch beliebig lange speichern, wenn man dafür Sorge trägt, daß zur Aufrechterhaltung der Ladungskapazitäten nach einem Zeitintervall $\Delta T$ eine Wiederaufladung (*Refresh*) zu erfolgen hat. Statische Halbleiterspeicher kennt man sowohl als Bipolar- wie auch als Unipolarbausteine. Dynamische Halbleiterspeicher existieren nur als Unipolarbausteine.

### 1.1 Statische Schreib-Lese-Speicher

Die Funktionsweise eines statischen Schreib-Lese-Speichers als Bipolarbaustein sei am Beispiel des elementaren RS-Flipflops erklärt, dessen S- bzw. R-Vorwahleingänge und dessen Ausgang durch zusätzliche Logik beschaltet sind (**Bild 2**). Um Daten einzuschreiben oder auszulesen, muß der Adreßeingang X mit H-Signal belegt sein. Beim Einlesen von Daten kann das am W-Eingang anliegende Signal nur dann wirksam werden, wenn zusätzlich der Freigabeeingang T aktiviert wurde. Liegt am W-Eingang (*Write*) H (L)-Signal an, liegt am Vorbereitungseingang S H (L)-Signal und am Vorbereitungseingang R L (H)-Signal an. Entsprechend tritt am Ausgang Q H (L)-Signal auf, d.h. das Signal entspricht dem über den W-Eingang eingeschriebenen Signal.

In der integrierten Schaltungstechnik lassen sich derzeit bis zu 16 384 solcher Speicherzellen einschließlich der notwendigen Steuerelemente auf einer Größe von 16 mm$^2$ unterbringen. In naher Zukunft werden von vielen Halbleiterherstellern

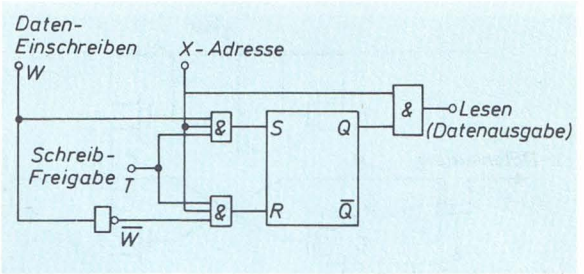



**Bild 2** Beschaltetes RS-Flipflop

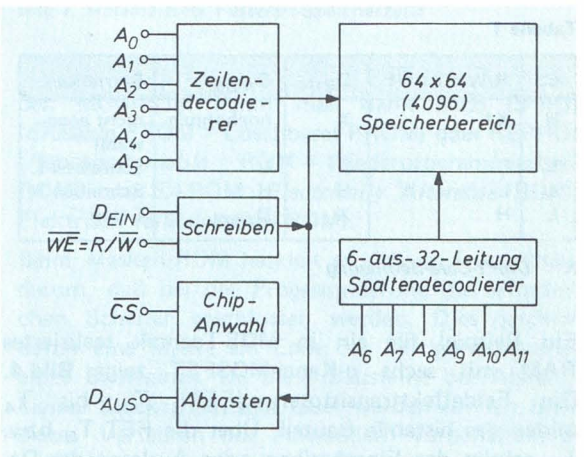



**Bild 3** Blockdiagramm eines 4 Kbit Schreib-Lesespeichers

RAM mit 65 536 bit Speicherkapazität angeboten werden. Zur Verdeutlichung der Organisation eines RAM mit 4096 bit = 4 Kbit sei das in **Bild 3** dargestellte vereinfachte Blockdiagramm gezeigt. Der Speicherbereich besteht aus einer 64reihigen Matrix (64 Zeilen und 64 Spalten). Zur Anwahl der entsprechenden Zeilen- bzw. Spaltenadressen benötigt man einen Zeilen- bzw. Spaltendecodierer mit je 6 Eingängen ($2^6 = 64$). Der Eingang $D_{EIN}$ entspricht dem W-Eingang von Bild 2. Der Eingang $\overline{WE} = R/\overline{W}$ dient zur Freigabe des Einschreibens bzw. Auslesens einer Information. $R/\overline{W}$ bedeutet „Lesen aber nicht Schreiben" (*Read but not Write*), d.h. der Speicherinhalt wird gelesen, wenn $R/\overline{W}$ = H. Es kann eingeschrieben werden, wenn $R/\overline{W}$ = L. Der Eingang $\overline{CS}$ dient zur Freigabe der Bausteine (CS: *Chip Select*). Das Zusammenwirken der einzelnen Funktionen ist noch einmal in der nachfolgenden Tabelle I zusammengefaßt.

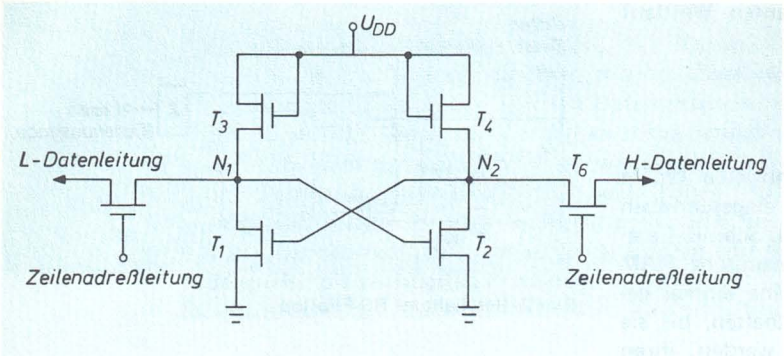



**Bild 4**
Speicherzelle mit sechs *n*-Kanal-MOS-FET

**Tabelle 1**

| $\overline{CS}$ | R/$\overline{W}$ oder $\overline{WE}$ | $D_{EIN}$ | $D_{AUS}$ | Betriebsart |
|---|---|---|---|---|
| H | X* | X | hochohmig | nicht angewählt |
| L | L | L | L | Schreiben L |
| L | L | H | H | Schreiben H |
| L | H | X | $D_{AUS}$ | Lesen |

X*: *Don't-Care-Bedingung*

Ein Beispiel für ein in MOS-Technik realisiertes RAM mit sechs *n*-Kanal-MOSFET zeigt **Bild 4.** Die Feldeffekttransistoren (FET) $T_1$ bis $T_4$ bilden das bistabile Bauteil. Über die FET $T_5$ bzw. $T_6$ erfolgt das Einschreiben oder Auslesen der Daten.

## 1.2 Dynamische Schreib-Lese-Speicher (RAM)

Dynamische RAM haben gegenüber statischen RAM den Vorteil, einen sehr viel kleineren Platzbedarf zu beanspruchen, da man die voluminösen Feldeffekttransistoren durch Kondensatoren ersetzen kann. Ein Beispiel für eine dynamische Speicherzelle, die aus einem Feldeffekttransistor und einem Kondensator besteht, zeigt **Bild 5.** Das Einschreiben einer Information erfolgt, indem man die Spannung, die an der Datenleitung anliegt, über den Feldeffekttransistor dem Kondensator zuführt. Das Lesen einer Information erfolgt so, daß sie in Abhängigkeit von der Kondensatorladung als Signalsprung auf der Datenleitung erscheint, wenn der Feldeffekttransistor T über die Zeilenauswahlleitung in leitenden Zustand versetzt wird. Der Nachteil dieser einfachen Speicherzelle liegt darin, daß der Speicherinhalt beim Auslesen verloren ist. Deshalb muß nach jedem Lesevorgang die verlorengegangene Information wieder eingespeichert werden. Aufgrund unvermeidlicher Leckströme, die temperaturabhängig sind,

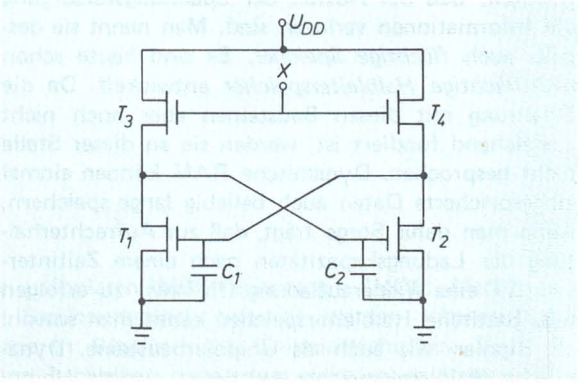



**Bild 5** Dynamische Schreib-Lese-Speicherzelle mit vier FET und zwei Kondensatoren

entlädt sich der Kondensator im Verlaufe der Zeit. Deshalb muß die gespeicherte Information (Kondensatorladung) in der Regel alle 2 ms regeneriert (*refreshed*) werden.

Heute werden vorzügliche Speicherzellen mit drei Feldeffekttransistoren verwendet (**Bild 6**). Soll in die Speicherzelle z.B. ein H-Signal eingeschrieben werden, gibt man auf die Schreibleitung ein H-Signal. Der Feldeffekttransistor $T_3$ schaltet durch. Liegt an der Daten/Lese-Leitung H-Signal an, lädt sich der Kondensator C auf. Nach Speicherung der Information wird auf die Schreibleitung wieder ein L-Signal gegeben. Der Feldeffekttransistor $T_1$ wird bei aufgeladenem Kondensator leitend. Beim Lesevorgang liegt sowohl an der Lese-Leitung als auch an der Daten/Lese-Leitung H-Signal an. Der Feldeffekttransistor $T_2$ wird leitend, so daß ein Strom durch $T_1$ und $T_2$ fließen kann. Das Signal auf der Lese-Leitung wird zu einem L-Signal. D.h. man erhält ein negiertes Speichersignal am Ausgang. Bei Einspeichern eines L-Signales hat die Daten/Lese-Leitung ein L-Signal.

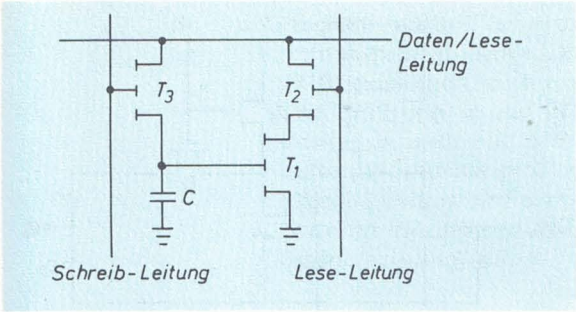



**Bild 6** Dynamische Schreib-Lese-Speicherzelle mit nur drei FET und einem Kondensator

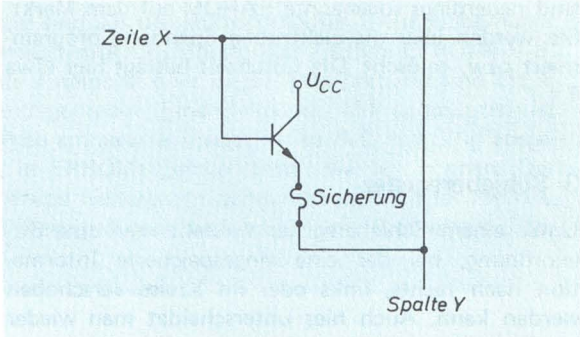



**Bild 7** Schema einer Festwert-Speicherzelle

Die Organisation von dynamischen RAM ist die gleiche wie bei den statischen RAM. Wichtige Kenngrößen bei RAM sind die Speicherkapazität, die Kosten pro Bit, die Abmessungen, der Leistungsverbrauch, die Zugriffs- und die Zykluszeit. Von den Kosten kann man sagen, daß sie bei höherer Packungsdichte abnehmen (etwa DM 0,03 bei 1024-Bit-RAM und etwa DM 0,002 bei 16 384-Bit-RAM). Die größten dynamischen RAM haben heute 65 536 Speicherzellen.

## 2 Festwertspeicher (ROM)

Ein ROM ist ein programmierter Festwertspeicher. Im Gegensatz zum RAM können in das ROM keine Informationen eingeschrieben, sondern nur ausgelesen werden. D.h. ein einmal in das ROM eingeschriebenes Programm kann nicht mehr geändert werden (diese Aussage wird weiter unten präzisiert). Dies hat allerdings auch den Vorteil, daß bei Stromausfall die Information (im Gegensatz zum RAM) nicht verloren geht und nach Wiederauftreten der Spannung wieder zur Verfügung steht. Im Prinzip handelt es sich beim ROM darum, daß man offene Schalter je nach Anforderung des auszuführenden Programmes schließt. Sind die Schalter geschlossen, können sie in der Regel nicht mehr geöffnet werden. Ein Beispiel für ein ROM mit geöffneten Schaltern (Sicherung leitend) zeigt **Bild 7**.

Man kennt bipolare und unipolare oder MOS-ROM. Sie unterscheiden sich im wesentlichen durch ihre Zugriffszeit; bipolare ROMs haben eine Zugriffszeit von etwa 50 ns und MOS-ROM von etwa 500 ns. Bei einer feineren Unterteilung unterscheidet man noch einmal zwischen Masken-ROM und PROM (*Programmable* ROM, Programmierbares ROM). Bei den MOS-ROM kennt man dann noch EPROM (*Erasable* PROM = Löschbares PROM) oder REPROM (*Reprogrammable* ROM = Wiederprogrammierbares ROM) und EAROM (*Electrically Alterable* ROM = Elektrisch veränderbares ROM).

Beim Masken-ROM handelt es sich im wesentlichen darum, daß bei der Programmierung die erforderlichen Schalter geschlossen werden. Dies geschieht durch eine Maske am Ende des Fertigungsprozesses eines Bausteines. Da diese Bausteine bei Abnahme kleiner Stückzahlen sehr teuer werden können, bleibt dieses Verfahren nur Anwendern vorbehalten, die große Mengen solcher Bausteine in ihren Geräten verwenden. Viele Anwender wollen auch bei kleineren Chargen die fertigen Bausteine selbst programmieren. Man hat deshalb die PROM geschaffen. Sie lassen sich beim Anwender programmieren und beruhen hauptsächlich darauf, daß man mit Hilfe von elektrischem Strom geschlossene Schalter durchtrennt. Der Nachteil dieser Programmierung liegt darin, daß ein menschlicher logischer Fehler bei der Programmierung nicht mehr geändert werden kann. Bei MOS-ROM kann diese Schwierigkeit durch EPROM umgangen werden. Sie sind spezielle Einheiten, sogenannte FAMOS (*Floating-Gate Avalanche-Injection MOS*)-Einheiten, die so programmiert werden können, daß sie leitend werden. Durch Einwirkung von ultraviolettem Licht können die Einheiten wieder in den nichtleitenden, also in den gelöschten Zustand, überführt und neu programmiert werden. Daß EPROM die normalen PROM nicht abgelöst haben, ist in der hohen Preisdifferenz zwischen beiden Einheiten begründet.

EPROM haben außerdem den Nachteil, daß sie nur mit UV-Licht gelöscht werden können. Die Löschzeit liegt zwischen 10 min und 30 min. Für schnelle Änderungswünsche bezüglich der Programmierung

sind neuerdings sogenannte EAROM auf dem Markt. Sie werden über die elektrische Spannung programmiert bzw. gelöscht. Die Löschzeit beträgt hier etwa 1 s.

## 3 Schieberegister

Unter einem Schieberegister versteht man eine Bitanordnung, bei der eine eingespeicherte Information nach rechts, links oder im Kreise verschoben werden kann. Auch hier unterscheidet man wieder bipolare und unipolare Schieberegister. Die bipolaren Schieberegister werden meist aus Flipflops in TTL-Technik gebildet. Ein Rechtsschieberegister für vier Bit zeigt **Bild 8.** Nach jedem Taktimpuls hat sich die jeweilige Information um eine Stelle nach rechts verschoben. Die aus TTL-Flipflops hergestellten Schieberegister haben in integrierter Schaltungstechnik eine unvertretbar hohe Verlustleistung und verbrauchen bei der Integration relativ viel Material, so daß die Herstellungskosten sehr hoch wären. Man kennt deshalb nur hochintegrierte Schieberegister (mit bis zu 65 536 Bit) in MOS-Technik. Für die technische Realisierung dieser Schieberegister kennt man verschiedene Möglichkeiten. Eine Realisierung zeigt **Bild 9**, das eine Schieberegisterstufe in *n*-Kanal-MOS-Technik (sechs MOSFET) zeigt. An Eingang $U_{Ein}$ liegt das Eingangssignal an. Zum Zeitpunkt $t = t_1$ bilden bei ansteigender Flanke des Taktimpulses $\phi_1$ die Feldeffekttransistoren $T_1$ und $T_2$ eine Inverterstufe. Der als Schalter wirkende Transistor $T_3$ wird durchgeschaltet. Wegen der Inverterstufe wird die zum Kondensator $C_1$ inverse Ladung auf den Kondensator $C_2$ übertragen.

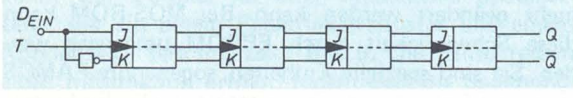



**Bild 8** 4 bit-Schieberegister

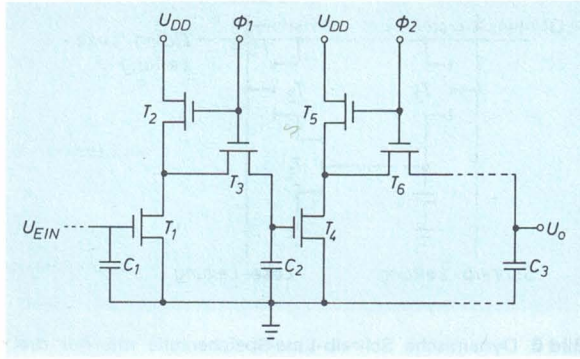



**Bild 9** Schieberegisterstufe in *n*-Kanal-MOS-Technik

Bei abfallender Flanke von Takt $\phi_1$ werden die Feldeffektransistoren $T_2$ und $T_3$ gesperrt; die Ladung auf Kondensator $C_2$ bleibt erhalten. Erfolgt anschliessend ein Taktimpuls $\phi_2$, wirken die Feldeffekttransistoren $T_4$ und $T_5$ als Inverter, d.h. der Feldeffekttransistor $T_6$ ist gesperrt. Dies bedeutet, daß die Ladung des Kondensators $C_2$ invertiert und an Kondensator $C_3$ weitergegeben wird. Dies wiederum bedeutet, daß die gespeicherte Information am Ausgang $U_0$, der gleichzeitig Eingang für die nächste Stufe ist, anliegt. Im Grunde hat man hier das aus der TTL-Technik bekannte *Master-Slave-Prinzip* verwirklicht.

Um Schieberegister mit extrem hoher Packungsdichte zu erhalten, arbeitet man mit Einheiten, die Ladungspakete bei Spannungsänderung verschieben. Man nennt sie auch CTD (*Charge Transfer Device*, Ladungsverschiebeeinheit) oder CCD (*Charge Coupled Device,* ladungsgekoppelte Einheit). Bei ihnen werden durch kapazitive Effekte Ladungspakete erzeugt, die durch Spannungsänderung an den einzelnen Anschlüssen von einem auf den nächsten Kondensator übertragen werden. Auf diese Weise können Informationen durch Ladungsverschiebung weitergeleitet werden.

Programmierbare Festwertspeicher sind aus der Mikrocomputer-Technik überhaupt nicht mehr wegzudenken. Es werden heute z.B. Monitore in 2 Kbyte EPROMs (z.B. 2716) gespeichert, auch residente Assembler oder BASIC-Interpretierer sind oft in acht 2708 oder in vier 2716 untergebracht. Eine Reihe von Mikrocomputern ist inzwischen auch mit EPROM-Programmern ausgestattet, so daß endgültig ausgetestete Programme problemlos in EPROMs „eingebrannt" werden können. Der folgende Beitrag gibt eine Reihe von weiteren Anwendungsbeispielen für PROMs, bespricht verschiedene PROM-Typen und geht ausführlich auf „Probleme bei der Programmierung" ein.

W. Link

# Programmierbare Festwertspeicher (PROM)

## 1 Entwicklung und Anwendung von PROMs

Ende der Sechziger Jahre kamen die ersten programmierbaren Halbleiter-Festwertspeicher (PROM) auf den Markt. Mit der Programmierbarkeit war eines der größten Hindernisse für eine weite Verbreitung von Halbleiter-Festwertspeichern beseitigt, nämlich das für die ROM-Herstellung erforderliche aufwendige und zeitraubende Verfahren – angefangen mit der Schreibtischarbeit für die Erstellung der benötigten Unterlagen bis zu den Wartezeiten bis zum Fertigungsbeginn, die oft bis zu zehn Wochen betrugen. Da für jede kundenspezifische Schaltung eigene **Masken** für die **IC-Herstellung** entworfen werden mußten, fielen hohe Maskenkosten an, so daß Rentabilität erst bei großen Stückzahlen erreicht wurde. Zeigten sich schließlich in der Testphase des fertigen Geräts Fehler im Programm bzw. wurden Verbesserungen erforderlich, so begann der ganze Zeit- und Kostenaufwand wieder von neuem – das ROM mitsamt Masken wurde unbrauchbar. Mit der Entwicklung der PROMs entfielen alle diese Schwierigkeiten. Der Anwender kann nun selbst in wenigen Minuten das gewünschte *Bitmuster* programmieren. Bei fehlerhafter Programmierung fallen lediglich die Kosten für ein neues IC an.

Seit Intel 1971 mit UV-löschbare und *wiederprogrammierbare* PROMs **(REPROMs)** auf den Markt brachte, wurde die Prototypen-Entwicklung noch einfacher; ein falsch programmierter IC konnte nach einem Löschvorgang wieder verwendet werden. Erst nach Beendigung der Testphase des REPROM-bestückten Geräts werden einfache PROMs programmiert, wobei das REPROM als Muster-PROM (*Master PROM*) dient.

Der Vorteil einfacher, zeitsparender und kostengünstiger Prototypen-Entwicklung mit PROMs war die eine, das enorme Anwachsen der Minicomputer-Industrie die andere Ursache für den steilen Umsatz-Zuwachs der PROM-Hersteller. Gerade hier bieten sich ROMs zur Speicherung von Programmen an, da letztere über längere Zeit beibehalten werden, eine Programmänderung also nur bei Änderung des Prozeßablaufs erforderlich wird. In der Groß-EDV finden Festwert-Speicher zunehmend weniger Verwendung, da man hier aus Gründen der Flexibilität immer mehr dazu übergeht, große Teile des Betriebssystems

**Masken:** Bei der Herstellung von integrierten Schaltkreisen (IC) werden die einzelnen Schaltelemente nacheinander auf das Substrat aufgedampft, indem entsprechende Masken jeweils den Rest abdecken.

**IC:** *Integrated Circuit*, d.h. Integrierte Schaltung (IS).

**REPROM:** Reprogrammierbare Festwertspeicher; heute als EPROM (*UV Erasable and Electrically Programmable ROM*) oder als EAROM (*Electrically Alterable ROM*) erhältlich. Wichtigste Typen: EPROM 2708 und 2716.

auf Platte bzw. **Diskette** ‚auszulagern' und vor Arbeitsbeginn wieder in die *Schreib-Lese-Speicher* (RAM) des Computers umzuladen.

Der in den vergangenen Jahren stark zunehmende Einsatz von *Mikroprozessoren* hat auch den PROMs viele neue Einsatzgebiete gebracht. So arbeiten PROMs in Computer-Peripheriegeräten, Automobilen, Maschinensteuerungen, Radios und Fernsehern, Telefon-Anlagen usw.

Im folgenden sollen einige Anwendungen näher dargestellt werden. **Bild 1** zeigt einen *Code-Wandler*. Die Code-Wörter des Eingangs-Codes bilden die Adressen eines n-bit tiefen ROMs unter denen die jeweiligen entsprechenden Code-Wörter des Ausgangs-Codes abgespeichert sind.

*Meßwandler* (Temperaturfühler, Druckmesser usw.) haben oft den Nachteil, daß sie nicht genügend linear sind. Hier hilft ein ROM bei der Fehlerkorrektur. An seinen Adreß-Eingängen liegt in digitaler Form der fehlerhafte Meßwert; am Ausgang erscheint der korrigierte Wert **(Bild 2)**.

Die übliche Art der Multiplikation durch Addition und Stellenverschiebung ist für manche Spezial-Anwendungen zu langsam; hier kann man in einem ROM eine *Multiplikations-Tabelle* speichern **(Bild 3)**.

Grundsätzlich läßt sich fast jedes Schaltnetz bzw. Schaltwerk durch ein PROM ersetzen, wobei alle Eingangskombinationen der vorgegebenen **Wahrheitstabelle** die Adressen bilden unter denen im ROM die zugehörigen Wahrheitswerte der Schaltfunktion abgespeichert sind. Betrachtet man nur die einfachen Und- bzw. Oder-Gatter (TTL-ICs), so kann man durch einfache Rechnung zeigen, daß ein 1K-ROM ca. 16 bis 32 ICs ersparen kann. Nimmt man jedoch die höher integrierten ICs der 74er Serie hinzu, so kommt man auf einen Durchschnittsbedarf von 37 ROM-Bits pro **Gatterfunktion,** was einer Ersparnis von 10 ICs bei einem 8 K-ROM bedeutet.

## 2 PROM-Typen

Eine weitverbreitete Fehlmeinung ist, daß PROMs nur aus sicherungsähnlichen Leiterbahnen bestehen und daß Programmierung das Wegbrennen dieser Leiterbahnen bedeutet. Es gibt zwei Gruppen von PROMs: **MOS-** und **Bipolar**-PROMs. MOS-PROMs enthalten die Informationen in Form von gespeicherten Ladungen; sie sind löschbar und wieder programmierbar. Bipolare Baustein-Schaltkreise erhalten ihre Information durch eine nicht mehr rückgängig zu machende Veränderung der Stromwege im IC. **Bild 4** gibt eine Übersicht über die verschiedenen PROM-Typen. Im folgenden soll auf diese Typen kurz eingegangen werden:

### 2.1 Bipolar-PROMs

*a) Mit zerstörbaren Leiterbahnen* **(Fusible Link)**

Hier gibt es drei verschiedene Arten, die sich im Material der Leiterbahnen unterscheiden: Polykristallines Silizium (Intel), Nickel-Chrom-Bahnen (NiCr), oder Titan-Wolfram-Bahnen (TiW). Die typische **Zellenstruktur** dieser Bausteine zeigt **Bild 5.** Die Programmierung erfolgt durch Wegbrennen der Leiterbahn. Dazu muß durch geeignete Signale an der Zeilen- und Spaltenanwahl-Leitung (**Row** and **Column**) der Transistor in den leitenden Zustand gebracht werden und kurzzeitig ein starker Strom fließen.

*b) Mit zerstörbaren pn-Übergängen* **(AIM)**

Bei diesen PROMs werden pn-Übergänge durch *Lawinen-Durchbruch* zerstört. Dabei wandern Aluminium-Atome in die Grenzschicht und sorgen für einen dauerhaft niederohmigen Übergang (*Avalanche Induced Migration*). **Bild 6** zeigt den Zellenaufbau. Bei der Programmierung muß man den Strom so bemessen, daß der pn-Übergang Q2 kurzgeschlossen wird, Q1 aber intakt bleibt!

### 2.2 MOS-PROMs

*a) UV-löschbare FAMOS-PROMs*

**Bild 7** zeigt eine typische **FAMOS**-PROM-Zelle (p-Kanal). Dieser Zellenaufbau wurde von Intel ent-

**Diskette:** Markenname für Floppy Disk (Flexible Magnetplatte).

**Wahrheitstabelle:** Auflistung aller logischen Zusammenhänge zwischen Eingangs- und Ausgangssignalen von Digitalschaltungen (Beschreibung der Schaltfunktion).

**Gatterfunktion:** Zum Beispiel UND, ODER, NAND.

**MOS:** *Metal-Oxid-Semiconductor*; Standard-Schichtstruktur für Feldeffekttransistoren (MOSFET).

**Bipolar** wirken „gewöhnliche" Transistoren, während FET „unipolar" arbeiten.

**Fuse:** Sicherung

**Fusible Link:** Schmelzbare Verbindung.

**Zelle:** Hier Speicherzelle zur Speicherung eines Bit.

**Row:** Zeile

**Column:** Spalte (Säule).

**AIM:** *Avalanche Induced Migration,* d.h. Lawinen-induzierte Wanderung.

**FAMOS:** *Floating Gate Avalanche Injection MOS,* d.h. MOS-Struktur mit „schwimmendem" Gate und Lawinen-Injektion.

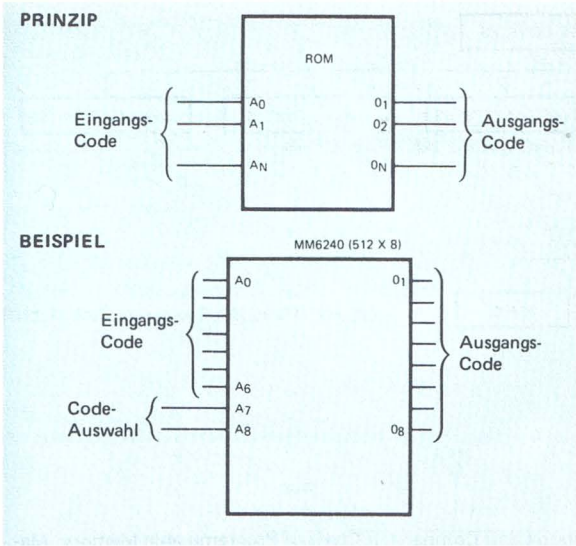
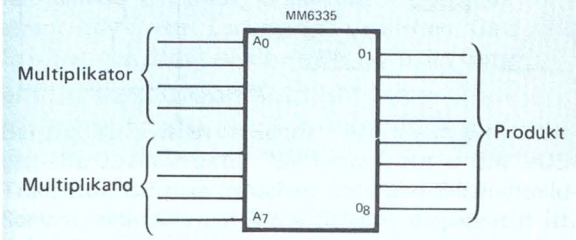



**Bild 1** Code-Wandler mit ROM

(Bilder 1 bis 3 sind entnommen aus *J. McDowell, L. Glas:* Bipolare Halbleiterspeicher, Monolithic Memories/Neumüller)

Aufgabe:
4 x 4 bit-Parallel-Binär-Multiplizierer
Prinzip:
Direkte Aufzeichnung des Produkts




Beispiel:
Multiplikator = 12 = 1 1 0 0
Multiplikand = 13 = 1 1 0 1
Produkt = 156 = 1 0 0 1 1 0 0

8 x 8-Multiplizierer

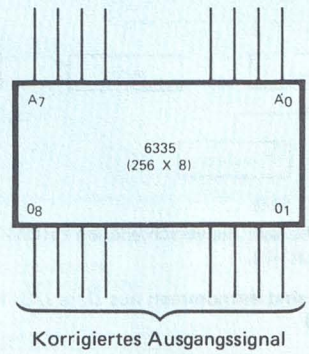
Die direkte Aufzeichnung erfordert einen ROM mit $2^{16}$ x 8 = 512 K bit, was sehr unpraktisch ist.

Lösung:
Man teile die 8 x 8-Multiplikation in eine 4 x 4-Multiplikation auf und addiere die Ergebnisse mit zwei TTL-4-bit-Addierer (74 S 181). Das erfordert nur einen 8 K-ROM.

**Bild 3** ROM als Multiplizierer

Aufgabe:
Bei einem digitalen Thermoumformer-Meßgerät sind folgende Fehler bekannt: +0,4 % Fehler zwischen −55 und 0 °C, +0,37 % Fehler zwischen 0 und 40 °C sowie die Fehlergleichung für den Bereich von 40 bis 125 °C: Fehler in % = (0,25 + Temperatur · 0,01−0,5). Die Fehlerkorrektur übernimmt der folgende Baustein:

Lösung: Ausgang des Thermoumformers




Durch Programmieren des PROM kann der Fehler korrigiert werden: Wenn beispielsweise am Ausgang das Signal 0 0 0 0 1 0 0 anliegt, das jedoch den Wert 0 0 0 0 1 0 1 haben müßte, dann muß bei der Binäradresse 4 der Wert 0 0 0 0 1 0 1 gespeichert werden.

**Bild 2** Fehlerkorrektur bei Meßwandlern

wickelt und 1971 zum ersten Mal auf den Markt gebracht. Die Zelle besteht aus einem MOS-Transistor, dessen **Gate**-Elektrode isoliert in Oxid eingebettet ist und keinen Kontakt zur übrigen Schaltung hat. Solange sich auf dieser Elektrode keine Ladung befindet, ist der Transistor gesperrt – der **Drain-Source**-Widerstand sehr groß. Zur Programmierung wird an die in Sperrichtung betriebene Drain-Zone eine hohe Spannung gelegt. Diese Spannung ruft einen *Lawinen-Durchbruch* hervor, wobei energiereiche, hochbeschleunigte Elektronen entstehen. Ein Teil von diesen durchdringt die Isolierschicht und gelangt auf die Gate-Elektrode **(Injektion).** Nach der Programmie-

**Gate, Drain, Source:** Anschlußelektroden eines MOS-Transistors.

**Injektion:** Einbringen von „Minoritätsträgern" in einen Halbleiter, also von Elektronen in p-Leiter bzw. von „Löchern" in n-Leiter.

**Injected Electrons:** Injizierte Elektronen, also Injektion von Elektronen in p-Leiter.

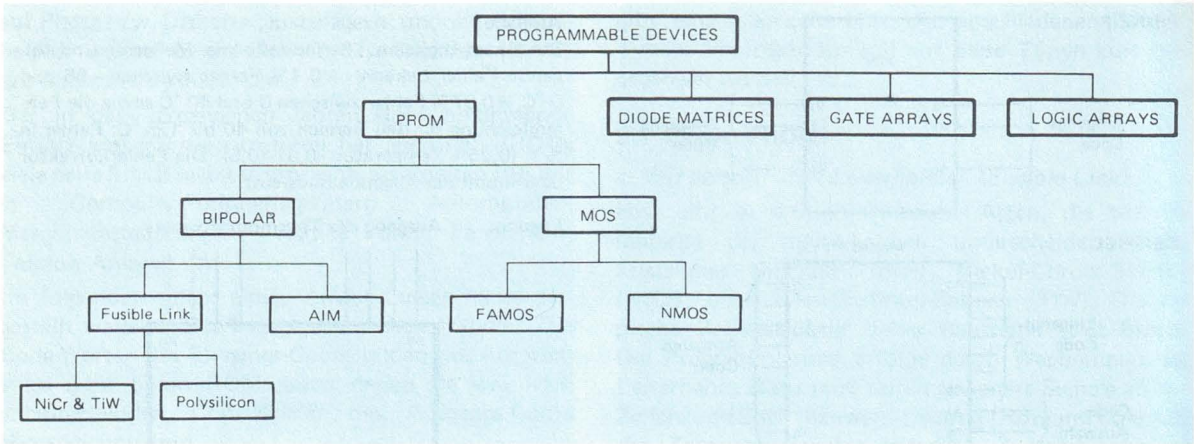



**Bild 4** Übersicht über die verschiedenen PROM-Typen

(Bilder 4 bis 8 sind entnommen aus *Data I/O:* Programming Handbook and Comparison Chart of Programmable Memory, Macrotron GmbH)

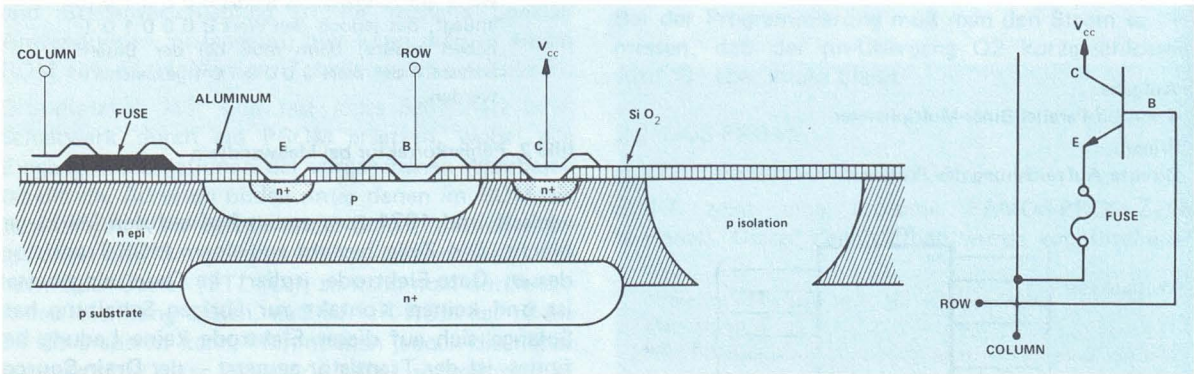



**Bild 5** ROM-Zelle mit zerstörbaren Leiterbahnen

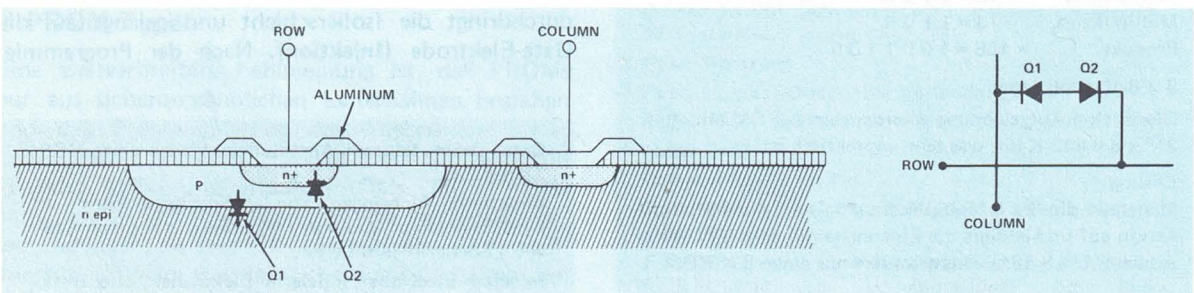



**Bild 6** Aufbau einer AIM-PROM-Zelle

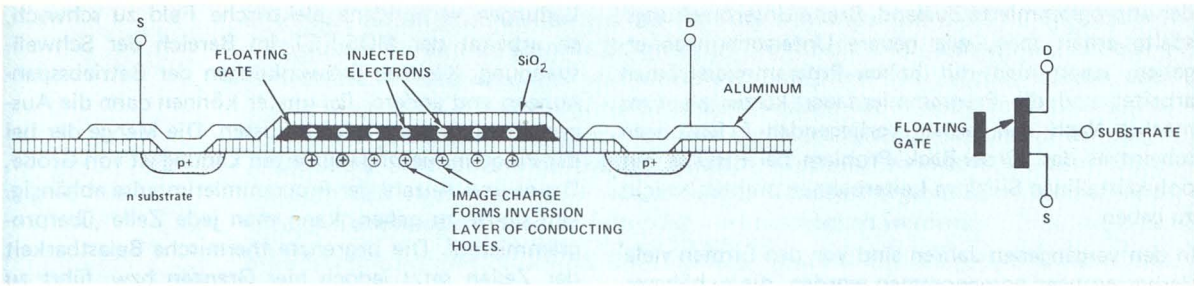



**Bild 7** Aufbau einer FAMOS-PROM-Zelle

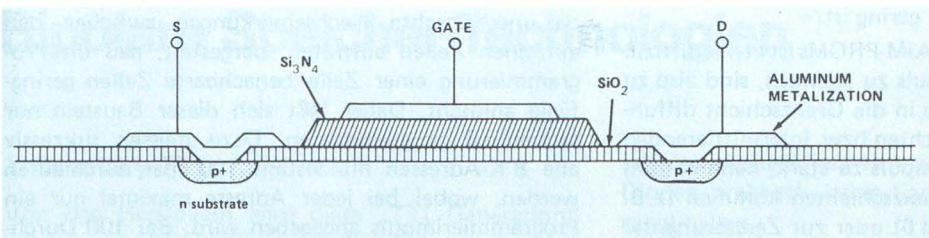



**Bild 8**
Aufbau einer MNOS-PROM-Zelle

rung binden die Gate-Elektronen durch **Influenz** positive Ladungen im Halbleitermaterial und bilden so einen dauerhaft leitfähigen Kanal zwischen Source- und Drain-Elektrode. Setzt man diese Zellen energiereichem UV-Licht aus, so führt der entstehende **Fotostrom** zu einem Ladungsfluß zwischen Gate und Substrat und damit zu einem Entladen des Gate.

*b) Elektrisch-löschbare MNOS-PROMs*

Bei **MNOS**-PROMs verwendet man die in **Bild 8** dargestellte Zellenstruktur. Sie besteht aus einem MOS-Transistor, bei dem zwischen Gate und Siliziumoxid-Schicht eine *Silizium-Nitrid-Schicht* eingebracht ist. Beim Programmieren werden durch Anlegen einer hohen Spannung an die Gate-Elektrode Ladungen in das Silizium-Nitrid *injiziert*, die wie bei der FAMOS-Zelle zur Bildung eines dauerhaft leitenden Kanals führen. Legt man nun eine hohe Spannung entgegengesetzter Polarität an die Gate-Elektrode, so werden die gebundenen Ladungen im Nitrid neutralisiert und damit das PROM gelöscht.

## 3 Probleme bei der Programmierung von PROMs

Die Programmierung bipolarer PROMs geschieht in einem Arbeitsgang Bit für Bit. Der Stromimpuls muß nach Stärke und Dauer so bemessen sein, daß möglichst nach dem ersten Impuls die Programmierung einer Zelle beendet ist. Die dazu benötigte Zeit beträgt i.a. einige Millisekunden.

Obwohl die PROM-Hersteller die in umfangreichen Untersuchungen ermittelten optimalen Werte für die Pulsbreite, Anstiegszeit und Anzahl der Programmier-Impulse angeben, treten gelegentlich beim Programmieren noch Fehler auf. In der Vergangenheit hatte man besonders bei TiW- und NiCr-Leiterbahnen mit dem **‚Grow Back'**-Phänomen zu kämpfen. Wenn nämlich der bei der Programmierung in der Leiterbahn entstehende Zwischenraum nur sehr klein ist, lagern sich unter dem Einfluß der zwischen den beiden Leiterteilen entstehenden hohen elektrischen Feldstärke (bis $10^7$ V/cm) Atome des Leitermaterials im Zwischenraum an. So entsteht mit der Zeit wieder

**Image Charge:** Bildladung.

**Influenz:** Ladungstrennung in einem Leiter, z.B. durch Annäherung eines geladenen Materials.

**Fotostrom:** Elektrischer Strom, der durch Ladungserzeugung bei Lichteinstrahlung entsteht.

**MNOS:** *Metal Nitride Oxide Semiconductor*, d.h. Metall-Nitrid-Oxid-Halbleiter, also MOS-Struktur mit zusätzlichem Isolator zwischen Metall und Oxid.

**Grow Back:** Rückbildung.

der unprogrammierte Zustand. Breite Unterbrechungsspalte erhält man, wie neuere Untersuchungen ergaben, wenn man mit hohen Programmierströmen arbeitet und die Programmierdauer kürzer als 1 ms macht. Nach den bisher vorliegenden Erfahrungen scheint es das ‚Grow-Back'-Problem bei PROMs mit polykristallinen Silizium-Leiterbahnen praktisch nicht zu geben.

In den vergangenen Jahren sind von den Firmen viele Verbesserungen vorgenommen worden, die zu höherer Zuverlässigkeit der programmierten Bausteine führten. Entgegen manchen optimistischen Herstellerangaben ist das Problem bis heute jedoch noch nicht vollständig gelöst, wenn auch der Ausschuß, verglichen mit dem früherer Jahre, heute sehr gering ist.

Die Programmierung der AIM-PROMs ist etwas diffizil. War der Programmierimpuls zu schwach, sind also zu wenig Aluminium-Atome in die Grenzschicht diffundiert, kommt es zu schlechten bzw. intermittierenden Verbindungen. Ist der Impuls zu stark, kann es zum Durchbruch weiterer Grenzschichten kommen (z.B. der Übergang Q1 in **Bild 6**) oder zur Zerstörung der Decodierlogik, die die Zeilen und Spaltenleitungen ansteuert. Von Vorteil bei den AIM-PROMs ist, daß man nach jedem Programmierimpuls den Widerstand der Zelle messen kann, während man bei den oben genannten PROMs nur feststellen kann, daß eine Unterbrechung der Leiterbahn besteht; wie groß der entstandene Spalt ist, läßt sich nicht angeben.

Bei den MOS-PROMs besteht das Problem der Zuverlässigkeit darin, wie lange sich die programmierte Information hält, wie lange sich also die injizierten Ladungen speichern lassen. Einwandfrei programmierte PROMs halten ihre Information über Jahre hinaus. Optimistische Angaben sprechen von 10 % Ladungsverlust in 10 Jahren.

Wichtig ist eine einwandfreie Programmierung jeder einzelnen Zelle. Ist das mit den injizierten Gate-Ladungen verbundene elektrische Feld zu schwach, so arbeitet der MOS-FET im Bereich der Schwellspannung. Kleinste Schwankungen der Betriebsspannungen und anderer Parameter können dann die Ausgangsspannung stark beeinflussen. Die Menge der bei der Programmierung injizierten Ladung ist von Größe, Dauer und Anzahl der Programmierimpulse abhängig. Um sicher zu gehen, kann man jede Zelle ‚überprogrammieren'. Die begrenzte thermische Belastbarkeit der Zellen setzt jedoch hier Grenzen bzw. führt zu übermäßig langen Programmierzeiten (früher z.T. bis 20 Minuten pro Baustein).

Bei den zwischenzeitlich erhältlichen 8 K-PROMs (2708 u.a.) ist dieses Verfahren kaum anwendbar, da unerwünschte Wechselwirkungen zwischen den einzelnen Zellen auftreten, dergestalt, daß die Programmierung einer Zelle benachbarte Zellen geringfügig anlöscht. Daher läßt sich dieser Baustein nur sequentiell programmieren. Dazu müssen sukzessiv alle 8 K-Adressen mindestens 100 mal durchlaufen werden, wobei bei jeder Adresse maximal nur ein Programmierimpuls abgegeben wird. Bei 100 Durchläufen (Programmier-Zyklen) ergibt das eine Gesamt-Programmierzeit von ca. 104 Sekunden.

Optimale Programmierzeiten erhält man, wenn man nach jedem Programmierimpuls- bzw. -zyklus beim 2708 – die Ausgangsspannung der Zellen oder den Source-Drain Widerstand mißt. Ein weiteres Problem tritt bei den MOS-PROMs auf. Zwar müßte ein solcher Baustein beliebig oft löschbar und wiederprogrammierbar sein. Leider steigt jedoch die zur Programmierung benötigte Zeit schon nach dreißig Neu-Programmierungen merklich an, so daß sich eine Maximalzahl von ca. 100 Lösch-Programmierzyklen ergibt. In diesem Zusammenhang muß daraufhingewiesen werden, daß eine stets einwandfreie Löschung der Bausteine für eine zuverlässige Neu-Programmierung unbedingt erforderlich ist.

In den vergangenen Jahren war für bipolare Anwendungen die TTL-Technik der Standard, für extrem schnelle Schaltungen (15 ns) die ECL-Technik. Zur Erzielung höchster Integrationsgrade und einfachster integrierter Strukturen wurden aber überwiegend MOS-Techniken herangezogen. Jedoch hat es nicht an Entwicklungsarbeiten gefehlt, mit denen die Vorteile der MOS-Technik mit den in der Regel größeren Schaltgeschwindigkeiten bipolarer Technik verknüpft wurden. Dabei haben sich verschiedenartige technologische Konzepte ergeben. Drei solcher bipolaren Technologien sollen im folgenden kurz besprochen werden.

H. Schumny

# Bipolare Speicher-Technologien *)

## 1 Schottky-TTL

Von den Herstellern wird diese „TTL-Generation" mit folgenden Argumenten angekündigt:

1. Höhere Geschwindigkeiten als bei Standard-TTL,
2. 80 % Verlustleistungseinsparung gegenüber Standard-TTL,
3. große Temperaturstabilität,
4. günstigere Preise.

Wegen Punkt 2 wird häufig von *Low-Power-Schottky-TTL* (LPS) gesprochen. Wie groß die Verlustleistungseinsparung pro Gatter ist, wird aus **Bild 1** deutlich. Lediglich für Schaltzeiten oberhalb 2,5 µs (unterhalb 0,4 MHz) kommen CMOS-Gatter mit geringerer Leistung aus.

Die Zelle eines Schottky-TTL-Gatters ist der sogenannte *Schottky-Transistor*. Wie aus **Bild 2a** zu ersehen, wird dieser Transistor realisiert, indem ein normaler bipolarer Transistor mit einer zwischen Basis und Kollektor geschalteten Schottky-Diode versehen wird. Es handelt sich bei diesem auch *Schottky-Barrier-Diode* (SBD) genannten Bauelement um einen Kontakt zwischen Metall und Silizium (sperrender Metall-Halbleiter-Übergang). In **Bild 2b** ist das vereinbarte Schaltbild angegeben. Der besondere Vorteil dieses neuen Bauteiles ist, daß mit nur etwa 0,3 V Spannungshub extrem hohe Schaltgeschwindigkeiten erzielt werden, weil Schottky-Dioden praktisch keine Speicherzeit (Schaltverzögerung) haben.

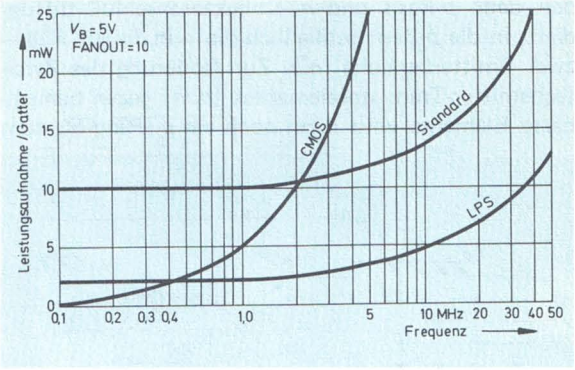



**Bild 1** Leistungsaufnahme in Abhängigkeit von der Schaltzeit bei einer Ausgangsfächerung (*Fan Out*) von 10 für CMOS, Standard TTL und LPS (*Low Power Schottky*)

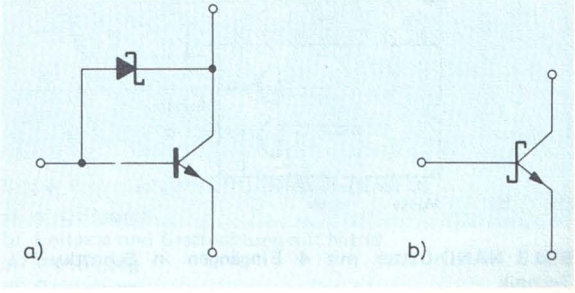



**Bild 2** Aufbau eines Schottky-Transistors (a) und Schaltbild (b)

*) Auszug aus *H. Schumny*, Digitale Datenverarbeitung für das technische Studium, Verlag Vieweg, Braunschweig/Wiesbaden.

Das Schaltbild eines Schottky-TTL-NAND-Gatters mit vier Eingängen zeigt **Bild 3a**. Zwei solcher Gatter sind in einem DIL-Gehäuse mit den Abmessungen 6,5 × 20 mm vereinigt **(Bild 3b)**. Höhere Packungsdichten sind leicht möglich, wobei der große Vorteil aller TTL-Glieder hervorgehoben sei, daß nur eine Versorgungsspannung $+U_B = 5\,V$ nötig ist (TTL-Pegel).

## 2 CDI- und BDI-Technologie

Hauptziele bei der Weiterentwicklung bipolarer Technologien sind die Erhöhung des Integrationsgrades (LSI = *Large Scale Integration*) sowie die Verringerung der Anzahl der Herstellungs-Prozeßschritte und der für Fotoätz- und Diffusionsschritte nötigen Masken. Zur Darstellung der in dieser Hinsicht erreichten Fortschritte sei mit **Bild 4** zunächst die bipolare TTL-Standardtechnik erläutert.

In ein *p*-leitendes Substrat wird danach zuerst ein $n^+$-Gebiet eindiffundiert – der spätere „vergrabene" Kollektor. Darauf wird epitaktisch eine *n*-Siliziumschicht aufgewachsen. In diese Epitaxieschicht werden dann *p*-Basis und $n^+$-Kollektoranschluß diffundiert, in die *p*-Basis schließlich die – in diesem Fall – zwei Emitterbereiche ($n^+$). Zur Isolierung des Zweifachemitter-Transistorelementes (*npn*) gegen benachbarte Elemente wird dann noch ein $p^+$-Ring bis zum Substrat hinab diffundiert. Für diese mit **Bild 4** beschriebene Standard-Technik (SBC = *Standard Buried Collector*) wurden 6 Masken für 4 Diffusions- und ebenfalls 4 Oxidationsschritte benötigt.

Eine Verringerung auf 5 Masken und je 3 Diffusions- und Oxidationsschritte wird mit der *CDI-Technik* möglich (CDI = *Collector Diffusion Isolation*). Wie aus **Bild 5** zu entnehmen, wird bei dieser Technologie

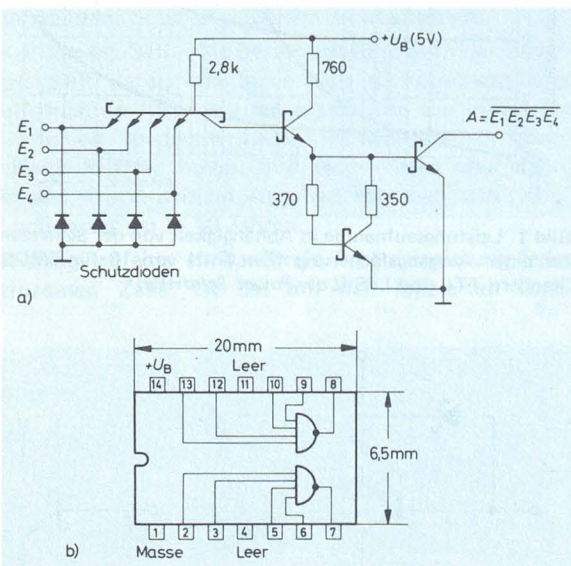



**Bild 3** NAND-Gatter mit 4 Eingängen in Schottky-TTL-Technik

a) Schaltschema
b) DIL-Gehäuse mit zwei Gattern

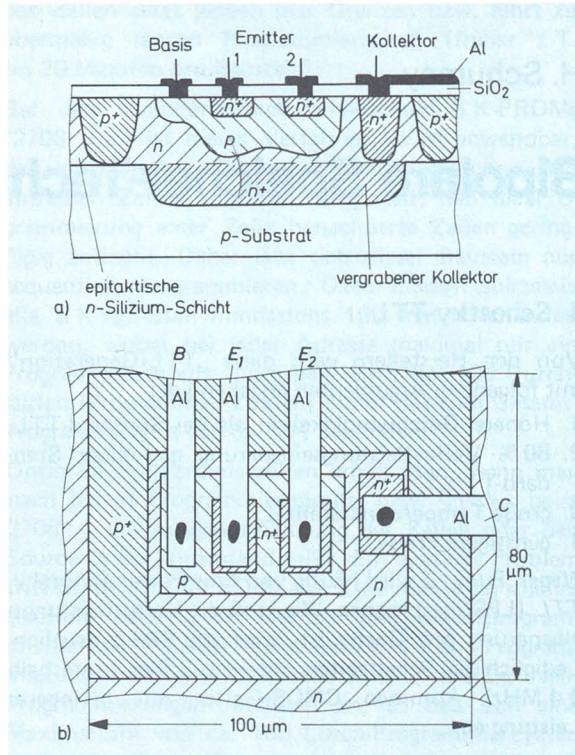



**Bild 4** Bipolare TTL-Standardtechnik mit „vergrabenem" Kollektor

a) Schematischer Querschnitt
b) Draufsicht (Durchsicht)

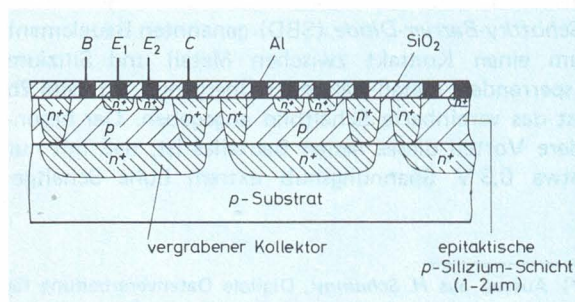



**Bild 5** *npn*-Transistorzellen mit 2 Emittern in CDI-Technik

auf ein $p$-Substrat eine $p$-Epitaxieschicht von 1 ... 2 $\mu$m Dicke aufgewachsen, die gleichzeitig als Basiszone und als Isolationsgebiet dient. Nun werden flache $n^+$-Zonen für die Emitter eindiffundiert, zuletzt die bis zu den vergrabenen Kollektoren reichenden $n^+$-Gebiete für den Kollektor-Anschluß. Diese $n^+$-Zonen begrenzen und isolieren gleichzeitig eine Transistorzelle, so daß auch der $p^+$-Schutzring entfallen kann. Neben der Verringerung der Maskenzahl und der Prozeßschritte wird so noch eine Integrationsdichte möglich, die der von MOS-Technologien entspricht.

Noch weniger Prozeßschritte erfordert die *BDI-Technik* (BDI = *Base Diffusion Isolation*). Dabei wird auf den vergrabenen Kollektor und die $p^+$-Isolationsdiffusion verzichtet. Aus **Bild 6** erkennt man, daß nun nur noch 4 Masken und je 2 Diffusion- und Oxidationsschritte nötig sind. Die $p^+$-Basis und der $p^+$-Schutzring werden nämlich gleichzeitig diffundiert.

Eine Gegenüberstellung des Aufwandes bei verschiedenen Technologien gibt **Bild 7.** Die Bedeutung der dort verwendeten Abkürzungen sei hier zusammengefaßt:

SBC Standard Buried Collector
CDI Collector Diffusion Isolation
BDI Base Diffusion Isolation
PMOS $p$-Channel MOS-Technique
MOSI MOS-Technik mit Ionen-Implantation
SGT Silicium Gate Technique
CMOS Complementary MOS
SOS Silicium On Sapphire-Technique

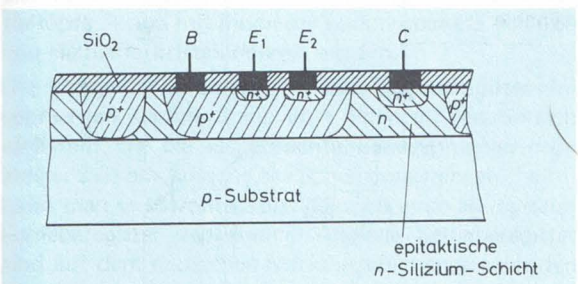



**Bild 6** *npn*-Transistorzelle in BDI-Technik

| Anzahl der | bipolar | | | unipolar | | | | |
|---|---|---|---|---|---|---|---|---|
| | SBC | CDI | BDI | PMOS | MOSI | SGT | CMOS | SOS |
| Masken | 6 | 5 | 4 | 4 | 4 | 4–5 | 6 | 6 |
| Diffusionen | 4 | 3 | 2 | 1 | 2–3 | 1 | 3 | 2 |
| Oxidationen | 4 | 3 | 2 | 2 | 2 | 3 | 3 | 2 |
| Epi-Schichten | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1–2 |

**Bild 7** Gegenüberstellung des Aufwandes bei verschiedenen Technologien

## 3 Isoplanar-Technik

Mit diesem Verfahren lassen sich ähnlich kleine Abmessungen erzielen wie mit der CDI-Technik. Wie in **Bild 8** gezeigt, wird auch hierbei in ein $p$-Substrat eine vergrabene Kollektorschicht diffundiert (*Buried Layer*). Darauf wird epitaktisch eine 2 $\mu$m dicke $p$-Schicht gewachsen. Der wesentlichste Unterschied zur CDI-Technik ist der, daß anstelle des isolierenden $n^+$-Ringes ein echter Isolator verwendet wird, nämlich das Siliziumoxid $SiO_2$. Zum Ätzen des isolierenden Grabennetzes und zum darauffolgenden Oxidieren wird eine maskierende Siliziumnitridschicht ($Si_3N_4$) verwendet **(Bild 8b und c).** Die geätzten Gräben werden dann bis zur Oberfläche der $p$-Epitaxieschicht oxidiert. Dieses Auffüllen bis zur Oberfläche hat zu

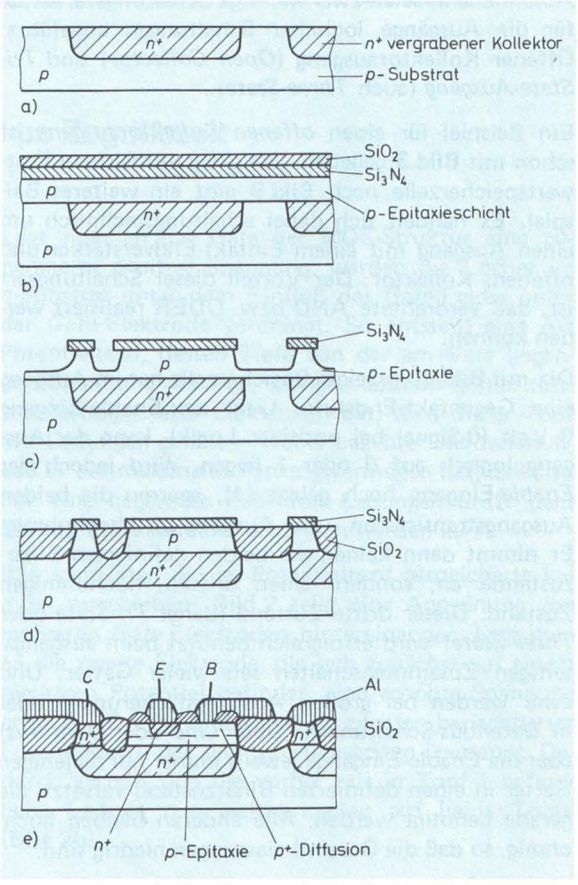



**Bild 8** Prozeßschritte der Isoplanartechnik
a) $n^+$-Diffusion
b) Epitaxie und Beschichtung mit Nitrid
c) Ätzvorgang
d) Oxidierung
e) nach Wegätzen des Nitrids folgen $n^+$-, $p^+$-Diffusion sowie weitere Oxidation und Al-Bedampfung

der Bezeichnung „isoplanar" geführt. Die in **Bild 8d** noch verbliebene $Si_3N_4$-Maske wird nun weggeätzt. Darauf folgt eine $p^+$-Diffusion für die Basis, $n^+$-Diffusionen für Emitter und Kollektor und Bedampfung mit Aluminium **(Bild 8e)**.

Während in der Standard-Technologie 4 Masken und 8 Prozeßschritte nötig sind (vgl. **Bilder 4 und 7)**, werden hier ebenfalls 4 Masken, aber nur 6 Prozeßschritte gebraucht. Die eigentliche Bedeutung dieses Verfahrens liegt jedoch in dem erheblich reduzierten Flächenaufwand. Denn Sicherheitsabstände sind nicht nötig, weil die Einzelsysteme durch einen echten Isolator getrennt werden. Die in **Bild 8e** angegebene $p^+$-Diffusion der Basis führt zu einer Erhöhung der Grenzfrequenz, also zu kürzeren Schaltzeiten.

Abschließend seien zwei wichtige Schaltungsvarianten für die Ausgänge logischer Schaltungen angeführt: Offener Kollektorausgang (*Open Collector*) und *Tri-State-Ausgang* (auch *Three-State*).

Ein Beispiel für einen *offenen Kollektorausgang* ist schon mit **Bild 3** gegeben. Die programmierbare Festwertspeicherzelle nach **Bild 9** gibt ein weiteres Beispiel. Es handelt sich dabei schaltungstechnisch um einen Ausgang mit einem Eintakt-Endverstärker und offenem Kollektor. Der Vorteil dieser Schaltungsart ist, daß verdrahtete AND bzw. ODER realisiert werden können.

Die mit **Bild 10** gezeigte Speicherzelle hat im Ausgang eine Gegentakt-Endstufe. Liegt am Enable-Eingang 0 Volt (0-Signal bei positiver Logik), kann der Ausgang logisch auf *0* oder *1* liegen. Wird jedoch der *Enable*-Eingang hoch gelegt (*1*), sperren die beiden Ausgangstransistoren – der Ausgang wird hochohmig. Er nimmt dann keinen der beiden definierten Binärzustände an, sondern einen dritten, hochohmigen Zustand. Dieser dritte Zustand (daher *Tri-State* oder *Three-State*) wird erfolgreich benutzt beim ausgangsseitigen Zusammenschalten sehr vieler Gatter. Und zwar werden bei großen Ausgangsfächerungen oder in Datenbus-Schaltungen (*Data Line* oder *Data Bus*) über die Enable-Eingänge jeweils immer nur diejenigen Gatter in einen definierten Binärzustand versetzt, die gerade benötigt werden. Alle anderen bleiben hochohmig, so daß die Gesamtbelastungen niedrig sind.

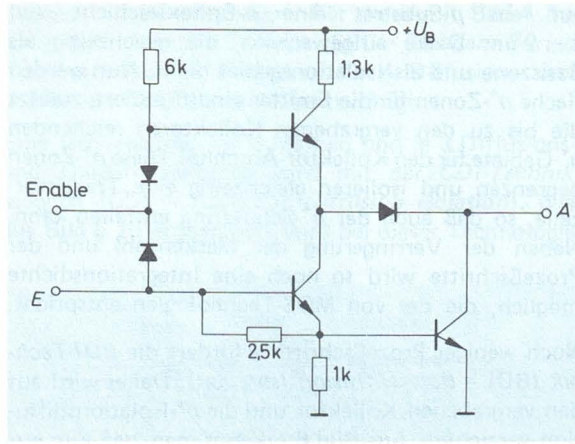



**Bild 9** Programmierbare Festwertspeicherzelle mit offenem Kollektor (*Open Collector*)

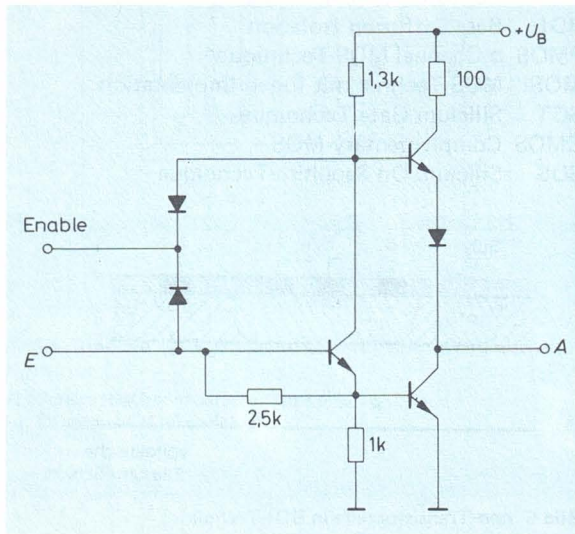



**Bild 10** Programmierbare Festwertspeicherzelle mit Tri-State-Ausgang

Die Stellung der magnetischen Massenspeicher in der Datenverarbeitung, – und zwar aller ihrer Varianten vom Magnetband bis zur Mini-Floppy – ist nicht mehr ganz unangetastet. In den Halbleiter-, Blasen- und Ladungsspeichern erwächst ihnen Konkurrenz. Wir stellen hier kurz einen solchen CCD-Speicher (mit extrem kurzer Zugriffszeit) vor, der nach außen hin einen Plattenspeicher „emuliert". Zuvor eine kurze Rekapitulation des CCD-Prinzips.

W. Link

# CCD-Halbleiterspeicher ersetzen rotierende Massenspeicher

## Ein CCD-Speicher für 1024 K Byte mit 250 µs Zugriffszeit

### 1 CCD-Speicher-Prinzip

Ladungsgekoppelte Speicherelemente (*Charge Coupled Devices*) sind im Prinzip *Schieberegister*, in denen Informationen in Form von elektrischen *Ladungspaketen* in *Potentialtöpfen* gespeichert werden. Diese Potentialtöpfe werden in MOS-ähnlichen Strukturen erzeugt. Unter dem Einfluß extern (an Steuerelektroden) angelegter Spannungen können die Potentialtöpfe – und mit ihnen die Ladungspakete – durch den Halbleiterkristall *bewegt* werden.

Die Menge der gesamten, in dieses Schieberegister eingebrachten Ladung kann über einen großen Bereich variieren. Da die eingebrachte Ladungsmenge nach einiger Zeit am Ausgang des Schieberegisters erscheint, kann man es sowohl als *analoges*, als auch als *digitales* Schieberegister verwenden. Analoge Schieberegister sind auf dem deutschen Markt erschienen, sie wurden als „Eimerkettenschaltungen" bezeichnet.

Der erste derartige ladungsgekoppelte Baustein wurde 1970 angekündigt. Inzwischen werden schon sehr komplexe Schaltkreise für eine Vielzahl von Anwendungen hergestellt, und weitere werden folgen. Man denke nur daran, daß man die zu verschiebende Ladung auch *fotoelektrisch*, also durch Lichteinstrahlung, erzeugen kann – woraus sich Anwendungen auf dem Gebiet der Optoelektronik ergeben.

Aus **Bild 1** ergibt sich die prinzipielle Funktionsweis eines CCD-Elements.

Legt man an das Gate der MOS-Struktur eine zunehmend positive Spannung, werden die Löcher im *p*-Substrat unter dem Einfluß des *Gate*-Feldes unter der *Gate*-Elektrode verdrängt. So entsteht eine Art Potentialtopf, dessen Tiefe von der am *Gate* liegenden Spannung abhängt. Bringt man nun Minioritätsladungsträger, also Elektronen ein, wird dieser Topf mit Ladungen gefüllt – wobei beachtet werden muß, daß er ein maximales Fassungsvermögen hat, daß also nur eine begrenzte Zahl von Ladungen unter dem Einfluß des Gatefeldes gebunden werden kann.

Wie kann man die im Potentialtopf gespeicherte Ladung verschieben? **Bild 2** zeigt eine Anordnung von mehreren *Gate*-Elektroden hintereinander. Legt man an die zweite Elektrode, die sich zunächst auf einem niedrigen Potential befindet, eine erhöhte Spannung an, z.B. 10 V, so bildet sich ein zweiter, benachbarter Potentialtopf, der sich mit dem ersten überlappt. Dadurch verteilt sich die vorher nur in Topf 1 befindliche Ladung nun gleichermaßen auf beide Töpfe **(Bild 2b)**.

Erniedrigt man die Spannung an der ersten Elektrode, so verschwindet der zugehörige Potentialtopf, und die darin gebundene Ladung fließt in den zweiten Topf. Damit ist die Ladung und die mit ihr verbundene Information um eine Stelle nach rechts gewandert. Legt man an die verschiedenen Elektroden ein entsprechendes *Impulsmuster* (Mehrphasentakt) an,

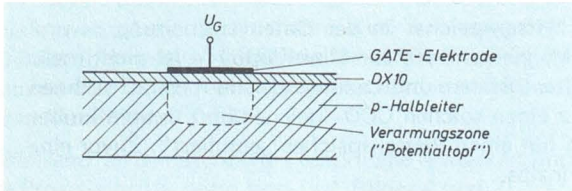



**Bild 1** Potentialtopf unter einer MOS-ähnlichen Struktur

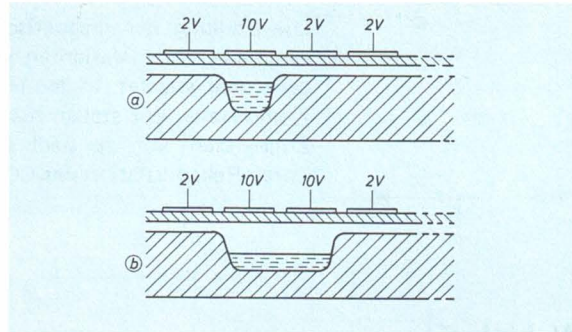



**Bild 2** CCD-Schieberegister. Mit Elektronen gefüllter Potentialtopf (a); die Ladung hat sich auf zwei „Töpfe" verteilt (b)

erhält man ein *Schieberegister*. Dabei fehlen lediglich noch die Elektroden für Eingabe und Ausgabe, die die zu verschiebenden Ladungen erzeugen und am Ende der Schiebekette wieder aufnehmen. Diese Elektroden bestehen aus zwei eindiffundierten, n-dotierten Inseln.

Zusammengefaßt kann man also sagen: CCD-Bausteine haben die gleiche technologische Basis wie MOS-Transistoren – aber sie verarbeiten Informationen nicht wie diese durch die Steuerung eines elektrischen Stromes, sondern durch die Verschiebung von kleinen Ladungspaketen. Die Speicherung der Daten aber geschieht im CCD-Speicher ebenso wie in dynamischen MOS-Speichern mit Hilfe von Ladungen.

Magnetische Speicher (Kern- und Blasenspeicher) benötigen relativ viel Energie. Elektro-mechanische Speicher benötigen zwar kleinere Netzteile zur Versorgung der Elektronik, jedoch zusätzliche Energie für den Antrieb der Motoren. Halbleiterspeicher benötigen weit weniger Energie als jeder der beiden erwähnten Speicher. So hat z.B. ein 1 Mbit-CCD-Speicher einen Leistungsverbrauch von weniger als 100 Watt.

## 2 Zugriffszeiten

Die mittleren Zugriffszeiten der bekannten Speicher bewegen sich zwischen Nanosekunden und Sekunden. Normalerweise kosten Speicher um so mehr, je kleiner die Zugriffszeit ist. *Bipolar-Schaltkreise* liegen am unteren Ende des Nanosekunden-Bereichs. An zweiter Stelle folgen *MOS-Schaltkreise*. Die langsamsten Vertreter dieser Gruppe sind etwa so schnell wie Kernspeicher – mit etwa 1 µs Zugriffszeit. *CCD-Schaltkreise* liegen im Bereich 100 µs bis 1 ms. *Blasenspeicher* werden etwa im 5-ms-Bereich liegen. *Plattenspeicher* und *Trommelspeicher* mit je einem Kopf pro Spur haben mittlere Zugriffszeiten von 5 bis 16,8 ms. *Plattenspeicher mit bweglichen Köpfen* solche von 40 bis 75 ms. *Floppy-Speicher* schließlich liegen bei einigen Zehntel-Sekunden. *Bandspeicher* haben die längsten Zugriffszeiten.

## 3 Speicherkosten

Obwohl CCD-Speicher erst sehr kurz auf dem Markt sind, liegen ihre Preise wesentlich *unter* denen älterer Bipolar- und MOS-Speicher bzw. Kernspeicher. CCD-Speicher sind im Augenblick viel teurer als ältere elektromechanische Speicher. Blasenspeicher liegen zwischen CCD und den elektromechanischen Speichern.

Der Preis von CCD-Speichern wird allerdings in dem Maße sinken, wie der Bedarf und damit die Produktion steigt und die Zahl der CCD-Lieferanten wächst. Zur Zeit besteht der Trend zur Entwicklung großer Speicher von 1 bis 4 MBit mit niedrigen Bit-Kosten.

CCD-Speicher haben den Nachteil, daß sie *flüchtige Speicher* sind – aber das sind MOS-Speicher schließlich auch. Die Industrie hat mittlerweile auch flüchtige MOS-Speicher als Arbeitsspeicher für Computer akzeptiert. Wo dies ein applikatives Problem ist, können CCD-Speicher mit einer zusätzlichen Batterie in einem nicht-flüchtigen Zustand gehalten werden.

## 4 Der CCDISC-Speicher

Die Firma *Alpha Data* hat auf der Basis von CCD-Bausteinen einen Halbleiter-Massenspeicher mit der Bezeichnung *CCDISC* entwickelt, der sich wie ein

Trommelspeicher bzw. ein Plattenspeicher mit je einem Kopf pro Spur verhält. Er ist also blockweise organisiert, mit wahlfreiem Zugriff und serieller Datenübertragung. Sein Kennzeichen ist die ultrakurze Zugriffszeit – verglichen mit dem Industrie-Standard 8,5 ms (bei Platten mit 3600 U/min). Er kann damit Plattenspeicher direkt ersetzen, in den Fällen in denen es um sehr kurze Zugriffszeiten geht.

Die mögliche Speicherkapazität bewegt sich dabei zwischen 128 KByte und 1024 KByte in einem einzelnen System (Modell SDM-1 bis SDM-8). Sie geht bis zu 4 Mega-Byte in einem Verbundsystem.

Als vorläufige technische Daten werden genannt:

- 500 µs maximale (250 µs mittlere) Zugriffszeit
- erweiterbar bis 1024 KByte in 128 K-Schritten
- 16 bis 128 „Spuren" sind adressierbar
- bis zu 4096 Worten (zu 16 bit) pro Spur (256 Worte pro Sektor)
- 2 bis 4 MHz Daten-Übertragungsrate Schreibschutz
- Warnung bei unzulässigen Adressen
- Eingebaute Spannungsversorgung Leistungsverbrauch 80 W
- Auf Wunsch zusätzliche Pufferbatterie zum Schutz gegen Spannungsausfälle.

**Bild 3** zeigt das Blockschaltbild der Speicherorganisation. Jede Speicherkarte nimmt 128 KByte auf.

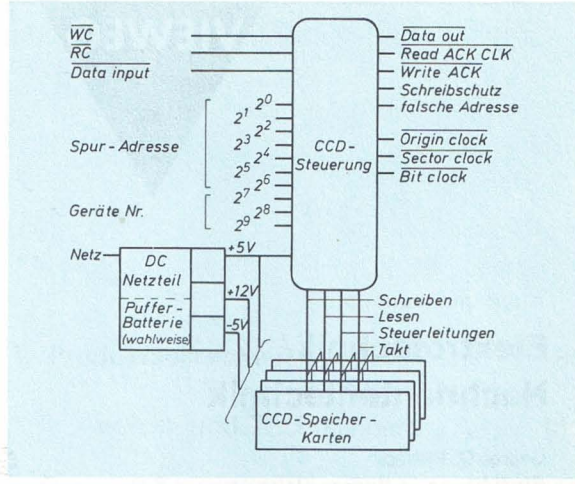
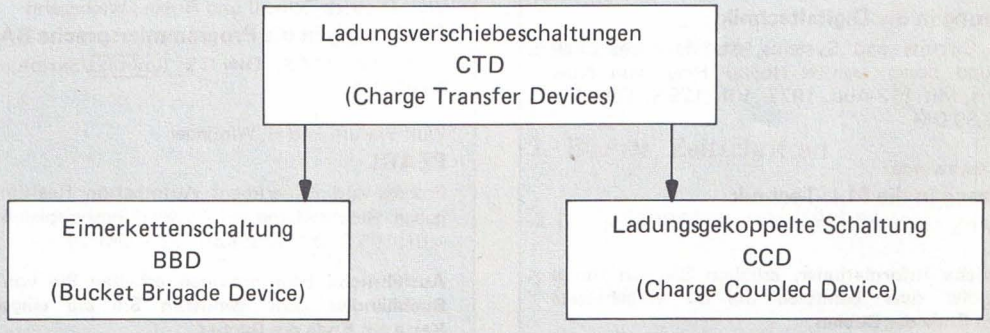



**Bild 3** Speicherorganisation des *CCDISC* von *alpha-data*

Das *CCDISC-Interface* ist steckkompatibel (TTL, decodiert) mit den Computer-Controllern desselben Herstellers, aber auch zu den eingeführten Minicomputer-Modellen anderer Firmen. Emulation der meisten Plattenspeicher ist möglich. Das System ist in ein 19-Zoll-Gehäuse (für Gestelleinbau) mit den Abmessungen 480 X 180 X 510 mm eingebaut.

**Arten von Ladungsverschiebeschaltungen**



VIEWEG

Aus dem Programm:

## Elektrotechnik/ Nachrichtentechnik

George D. Bishop
**Einführung in lineare elektronische Schaltungen**
(Linear Electronic Circuits and Systems, dt., aus dem Engl. übers. von Franz Seifert.) Hrsg. von Noel M. Morris. Mit 118 Abb. 1977. VIII. 151 S. DIN C 5. Kart. 19,80 DM.

Erwin Böhmer
**Elemente der angewandten Elektronik**
Kompendium für Ausbildung und Beruf. Mit 424 Abb. 1979. VIII, 283 S. 17 X 24 cm (Viewegs Fachbücher der Technik). Kart. 39,50 DM.

Das Buch gibt eine zusammenfassende Darstellung der für die angewandte Elektronik wesentlichen Bauelemente und Grundschaltungen. Es ist in 127 Sachthemen gegliedert und führt den Leser systematisch von der Darstellung einfacher Bauelemente und Schaltungen zu Systemen mit einem höheren Entwicklungsgrad.

Bodo Morgenstern
**Elektronik**
**Band 1:** Bauelemente. Mit 255 Abb. 1978. XIV, 153 S. DIN C 5 (uni-text/Skriptum). Pb. 24,80 DM.

**Band 2:** Schaltungen. Mit 231 Abb. 1978. XI, 267 S. DIN C 5 (uni-text/Skriptum). Pb. 24,80 DM.

Noel M. Morris
**Einführung in die Digitaltechnik**
(Digital Circuits and Systems, dt., aus dem Engl. übers. und bearb. von N. Hojka) Hrsg. von Noel M. Morris. Mit 112 Abb. 1977. VII, 129 S. DIN C 5. Kart. 17,80 DM.

Horst Geschwinde
**Einführung in die PLL-Technik**
Mit 80 Abb. 1978. IV, 117 S. DIN C 5. Kart. 15,80 DM.

**Ausführliche Informationen erhalten Sie von Ihrem Buchhändler oder benutzen Sie die eingeheftete Karte am Ende des Buches.**

VIEWEG

## Programmiersprachen

Hans Ackermann
**BASIC in der medizinischen Statistik**
1977. VIII, 114 S. DIN C 5 (uni-text/Skriptum.) Pb. 16,80 DM.

Harry Feldmann
**Einführung in ALGOL 68**
1978. IX, 311 S. DIN C 5 (uni-text/Skriptum.) Pb. 29,80 DM.

Günther Lamprecht
**Einführung in die Programmiersprache FORTRAN IV**
Eine Anleitung zum Selbstudium. Nachdr. d. 3. ber. Auf. 1976. IV, 194 S. DIN C 5 (uni-text/Skriptum.) Pb. 17,80 DM.

**Einführung in die Programmiersprache SIMULA**
Anleitung zum Selbstudium. Mit 41 Abb. u. 1 Tafel, 1976. IV, 231 S. DIN C 5 (uni-text/Skriptum.) Pb. 19,80 DM.

Wolfgang Schneider
**BASIC**
Einführung für Techniker. 2., durchges. Aufl. 1979. VI, 134 S. DIN C 5 (Viewegs Fachbücher der Technik). Kart. 19,80 DM.

**FORTRAN**
Einführung für Techniker. 1977. VI, 129 S. DIN C 5 (Viewegs Fachbücher der Technik). Kart. 19,80 DM.

Wolf-Dietrich Schwill und Roland Weibezahn
**Einführung in die Programmiersprache BASIC**
1976. IV, 114 S. DIN C 5 (uni-text/Skriptum.) Pb. 15,80 DM.

Wulf Werum und H. Windauer
**PEARL**
Process and Experiment Automation Realtime Language. Beschreibung mit Anwendungsbeispielen. 1978. VIII, 195 S. DIN C 5. Kart. 32,– DM.

**Ausführliche Informationen erhalten Sie von Ihrem Buchhändler oder benutzen Sie die eingeheftete Karte am Ende des Buches.**

# Daten

## Inhaltsübersicht


1. Produktübersichten . . . . . 186

1.1 Taschen- und Tischrechner . . . . . 186

1.2 Programmierbare Taschen- und Tischrechner . . . . . 211

1.3 Mikroprozessoren . . . . . 215

1.4 Mikrocomputer . . . . . 224

2. Adressen . . . . . 237

2.1 Taschen- und Tischrechner-Hersteller . . . . . 237

2.2 Mikroprozessoren-Hersteller . . . . . 239

2.3 Mikrocomputer-Hersteller . . . . . 240

2.4 Lieferanten (Distributoren) . . . . . 242

2.5 Clubs und Software-Lieferanten . . . 244

2.6 Lehrinstitute . . . . . 245

3. Bücher, Zeitschriften . . . . . 246

3.1 Taschenrechner-Bücher . . . . . 246

3.2 Mikrocomputer-Bücher . . . . . 247

3.3 Zeitschriften . . . . . 249

# 1 Produktübersichten

Die nachfolgenden Datentabellen sind so ausgeführt, daß ein schneller Vergleich der wesentlichen Eigenschaften verschiedener Taschenrechner, Mikroprozessoren etc. möglich wird. Um dies zu erreichen, wurde die Anzahl der Vergleichsdaten relativ gering gehalten. Dadurch ist sicher im Einzelfall keine vollständige Beurteilung durchführbar. Aber das kann auch gar nicht der Zweck umfassender Produktübersichten sein. Vielmehr ist denkbar, daß beispielsweise nach einer Vorauswahl anhand der Datentabellen für die dann vielleicht verbleibenden wenigen Geräte oder Bausteine detaillierte Herstellerangaben beschafft werden.

Um die Produktübersichten möglichst umfassend zu gestalten und damit gewissermaßen einen „Grundstock" für kommende Jahrbücher zu gewinnen, wurde vor allem bei Taschenrechnern, Mikroprozessoren und Mikrocomputern Wert auf Vollständigkeit gelegt, indem auch Typen und Geräte aufgenommen sind, die nicht mehr hergestellt und verkauft werden. Dadurch entsteht aber gerade der beabsichtigte systematische Überblick. In den folgenden Ausgaben des Jahrbuches sind dann nur noch kontinuierliche Fortschreibungen der technischen Entwicklung nötig.

## 1.1 Taschen- und Tischrechner

Die Datentabellen „Taschen- und Tischrechner" enthalten nur *nicht-programmierbare Rechner* und sind nach Herstellern alphabetisch geordnet. Sie reichen darum von „Addiator" bis „WALTHER". Diese Systematik wurde gewählt, weil Taschen- und Tischrechner in der Regel unter den Herstellernamen bekannt sind und beurteilt werden.

In der Spalte „Rechenlogik" werden folgende Abkürzungen benutzt:

AL – Algebraische Logik
AOS – Algebraisches Operationssystem
ARL – Arithmetische Logik
UPN – Umgekehrte Polnische Notation

Als „Funktionen" sind nur solche Tastenaufrufe gewertet, durch die eine Variable $x$ im mathematischen Sinne direkt beeinflußt wird. Also $\sqrt{x}$, $x^2$, $1/x$, $x!$, trigonometrische, logarithmische, exponentielle und statistische Funktionen.

Die Spalte „Anzeige" gibt ebenfalls mit Abkürzungen die Art an. Dabei bedeuten:

LED – *Light Emitting Diode* (Leuchtdiode; in der Regel rot)
LCD – *Liquid Crystal Display* (Flüssigkristallanzeige)
DIG – *Digitronic* (Fluoreszenzanzeige; in der Regel grün).

Bei den hier aufgelisteten nicht-programmierbaren Rechnern kommen in der Spalte „Ergänzungen" nur Drucker vor. Wenn es sich um einen Rechner ohne Anzeige handelt, ist zusätzlich die ausdruckbare Stellenzahl angegeben. Andernfalls entspricht sie der Stellenzahl der Anzeige.

Als Besonderheiten sind alle nicht-mathematischen Funktionen, Konstanten etc. angegeben. Dazu gehören auch %, +/–, M+, M– etc. Mit „Umwandlungen" sind solche zwischen verschiedenen Maßsystemen oder Zahlensystemen gemeint. Die manchmal aufgeführten Stundenzahlen bedeuten die mit einer Batteriefüllung möglichen Betriebsstunden.

*Eine Anmerkung:* Die japanische Taschen- und Tischrechnerproduktion ist nach wie vor von einer überwältigenden Fruchtbarkeit. Jedes Jahr wird der Markt mit einer wahren Flut von neuen Typen überschüttet. Bei genauerem Hinsehen fällt aber bald folgendes auf: Sehr oft sind es immer wieder die technisch gleichen Typen, die nur in neuer Aufmachung und mit anderer Bezeichnung als „Neu" vorgestellt werden. Andererseits scheint eine große Zahl der japanischen Rechner sich nur im Herstellernamen zu unterscheiden. So wird die in den nachfolgenden Datentabellen enthaltene Vielfalt oft nur vorgetäuscht.

| Hersteller Typ | Rechenlogik | Anzahl der Klammern | Anzahl der Funktionen | Anzahl der Tasten | Tastenbeleg. einfach | Tastenbeleg. mehrfach | Zahl der Speicher | Zahl der Stellen | Komma Fest- | Komma Fließ- | Komma Gleit- | techn.-wiss. | kaufmännisch | Anzeige Art | Anzeige Stellen | Stromversorgung Akku | Stromversorgung Batterie | Stromversorgung Netz | Ergänzungen | Besonderheiten |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **• Addiator** | | | | | | | | | | | | | | | | | | | | |
| EL82C | AL | | | 21 | X | | 1 | 8 | | X | | | X | DIG | 8 + 1 | | X | X | | %/+/_/Stoppuhr |
| EL84 | AL | | 1 | 22 | X | | 1 | 8 | | | | | | DIG | 8 + 1 | | X | X | | %/+/_ |
| **Adler** | | | | | | | | | | | | | | | | | | | | |
| Solar 1980 | AL | | 1 | 24 | X | | 1 | 8 | | X | | | | LCD | 8 + 1 | | | | | %/M+/M–/+/_/Solarzellen |
| **• ARISTO** | | | | | | | | | | | | | | | | | | | | |
| M16 | AL | | | 20 | X | | 1 | 8 | | X | | | | LED | 8 + 1 | | X | X | | % |
| M36 | AL | | | 20 | | X | 1 | 8 | | X | | | | LED | 8 + 1 | | X | X | | %/M+/M– |
| M42 | AL | | 1 | 25 | X | | 1 | 8 | | X | | | | DIG | 8 + 1 | | X | X | | %/M+/M– |
| M64 | AL | | 1 | 20 | | X | 1 | 8 | X | X | | | | LED | 8 + 1 | | X | X | | %/M+/M– |
| M65 | AL | | 3 | 20 | | X | 1 | 8 | X | X | | | | LED | 8 + 1 | | X | X | | %/M+/M–/+/_ |
| M66 | AL | | 3 | 25 | | X | 1 | 8 | | X | | | | DIG | 8 + 1 | | X | X | | %/M+/M–/π/+/_ |
| M75 | AL | | 13 | 20 | | X | 1 | 8 | | X | | X | | LED | 8 + 1 | | X | X | | π/400⁹/rad/M + x² |
| M76 | AL | | 13 | 20 | | X | 1 | 10 | | | X | X | | DIG | 8 + 1 | | X | X | | π/M+/M–/+/_ |
| M80 | AL | | 10 | 20 | | X | 1 | 8 | | | X | X | | LED | 8 + 1 | | X | X | | π/rad/M+/M– |
| M80S | AL | | 10 | 20 | | X | 1 | 8 | | | X | X | | LED | 8 + 1 | X | | | | π/rad/M+/M– |
| M85 | AL | 2 | 14 | 25 | | X | 1 | 10 | | | X | X | | LED | 10 + 2 | | X | X | | π/rad/M+/M–/Mx/M÷ |
| M85S | AL | 2 | 14 | 25 | | X | 1 | 10 | | | X | X | | LED | 10 + 2 | X | | | | π/rad/M+/M–/M x/M÷ |
| M400 | AL | | 1 | 24 | X | | 1 | 8 | | X | | | X | LCD | 8 + 1 | | X | | | %/M+/M–/+/–/2000 Stunden |
| M700 | AL | 2 | 21 | 35 | | X | 1 | 8 | | | X | X | | LCD | 8 + 1 | | X | | | π/M+/M–/Statistik |
| M800 | AL | 2 | 24 | 35 | | X | 1 | 10 | | | X | X | | LCD | 10 + 1 | | X | | | π/M+/M–/Hyperb./Kombinatorik/rad/Polarkoord./Statistik |
| Parat | AL | | | 20 | | X | 1 | 8 | | X | | | | LED | 8 + 1 | | X | X | | %/M+/M– |
| R4000 | ARL | | | 25 | X | | 1 | | X | X | | | X | DIG | 10 + 1 | | X | X | | %/Δ%/M+/M–/kleiner Tischrechner |
| Top | AL | | | 18 | X | | | | | X | | | X | LED | 8 + 1 | | X | X | | % |
| Tourist | AL | | 1 | 24 | X | | 1 | | | | | | | | | | | | | %/M+/M–/+/_/Permanentspeicher |
| **• Aviatronik** | | | | | | | | | | | | | | | | | | | | |
| Cristallonic 2000 | AL | | | 20 | X | | 1 | | X | X | | | | LED | 9 + 1 | X | | | | % |
| **• BMC** | | | | | | | | | | | | | | | | | | | | |
| 85 | AL | | | 18 | X | | | | | X | | | | DIG | 8 + 1 | | X | X | | % |
| 88 | AL | | 1 | 20 | X | | | | | X | | | | DIG | 8 + 1 | | X | | | % (≙ Mini-8) |
| 88M | AL | | | 24 | X | | 1 | 8 | | X | | | | DIG | 8 + 1 | | X | | | %/+/_/M+/M– (≙ Mini-8M) |
| 88P | AL | | 1 | 24 | | X | 1 | 8 | | X | | X | | DIG | 8 + 1 | | X | | | %/+/_/M+/M–/π (≙ Mini-8P) |
| 88S | AL | | 3 | 24 | | X | 1 | 8 | | X | | X | | DIG | 8 + 1 | | X | | | %/+/_/M+/M–/π (≙ Mini-8S) |
| 800PR | AL | | 1 | 20 | X | | | | | X | | | | DIG | 8 + 1 | | X | | | % |
| 801M | AL | | | 24 | X | | 1 | 8 | | X | | | | DIG | 8 + 1 | | X | | | %/+/_/M+/M– |
| 802MR | AL | | 3 | 24 | | X | 1 | 8 | | X | | X | | DIG | 8 + 1 | | X | | | %/+/_/M+/M–/π |
| 1010P | ARL | | | 28 | X | | 1 | 10 | X | X | | | X | | | | | X | Drucker (10 Stellen) | Tischrechner/%/+/_/M+/M– |
| 1200PD | ARL | | | 26 | X | | 1 | 12 | X | X | | | | DIG | 12 + 1 | | | X | Drucker | Tischrechner/M+/M–/% |
| 1212M | ARL | | | 23 | X | | 1 | 12 | X | X | | | | DIG | 12 + 1 | | X | X | | kleiner Tischrechner/M+/M– |
| 1212PDS | ARL | | | 32 | X | | 4 | 12 | X | X | | | X | DIG | 12 + 1 | | | X | Drucker | Tischrechner/%/Δ %/+/_/M+/M– |
| 8003M | AL | | 1 | 22 | X | | 1 | 8 | | X | | | | DIG | 8 + 1 | | X | X | | % |
| CPD-10 | AL | | | 26 | X | | 1 | 10 | X | X | | | | DIG | 10 + 1 | | | X | Drucker | Tischrechner/%/M+/M– |
| Credit Card Quartz | AL | | | 22 | X | | | | | X | | | | LCD | 8 + 1 | | X | | | Uhr, Kalender, Wecker, 10 000 Stunden |
| EC Alarm Clock | AL | | | 22 | X | | 1 | 8 | | X | | | | LCD | 8 + 1 | | X | | | %/+/_/M+/M–/wie CC Quartz |
| EC LCD | AL | | | 22 | X | | 1 | 8 | | X | | | | LCD | 8 + 1 | | X | | | %/M+/M–/2500 Stunden |
| EC Memo Save | AL | | | 22 | X | | 1 | 8 | | X | | | | LDC | 8 + 1 | | X | | | %/M+/M–/2 500 Stunden/Permanentspeicher |
| LC-10D | ARL | | 1 | 26 | X | | 1 | 10 | X | X | | | | LCD | 10 + 1 | | X | | | kleiner Tischrechner/%/Δ %/+/_/M+/M–/10 000 Stunden |
| LCD-8M | AL | | 1 | 24 | X | | 1 | 8 | | X | | | | LCD | 8 + 1 | | X | | | %/+/_/M+/M–/2 200 Stunden |
| LCD-8S | AL | | 18 | 36 | | X | 1 | 8 | | | X | X | | LCD | 8 + 1 | | X | | | π/+/_/M+/M–/rad |
| LCD-8W | AL | | 1 | 31 | | | | | | X | | | | LCD | 8 + 1 | | X | | | %/+/_/M+/M–/Wecker/Stoppuhr/Kalender/13 000 Stunden |
| MPD-10 | AL | | | 28 | X | | 1 | 10 | X | X | | | | DIG | 10 + 1 | X | | X | Drucker | Kleiner Tischrechner/%/+/_/M+/M– |
| MPD-10CT | ARL | | | 29 | X | | 1 | 10 | X | X | | | X | DIG | 10 + 1 | X | | X | Drucker | wie MPD-10 |

| Hersteller Typ | Rechenlogik | Anzahl der Klammern | Anzahl der Funktionen | Anzahl der Tasten | Tastenbeleg. einfach | Tastenbeleg. mehrfach | Zahl der Speicher | Zahl der Stellen | Komma Fest- | Komma Fließ- | Komma Gleit- | techn.-wiss. | kaufmännisch | Anzeige Art | Anzeige Stellen | Stromversorgung Akku | Stromversorgung Batterie | Stromversorgung Netz | Ergänzungen | Besonderheiten |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| noch | | | | | | | | | | | | | | | | | | | | |
| ● **BMC** | | | | | | | | | | | | | | | | | | | | |
| MPD-12S | AL | | | 29 | X | | 1 | 12 | X | X | | | X | DIG | 12 + 1 | X | | X | Drucker | wie MPD-10 |
| SR88 | AL | | 13 | 24 | | X | 1 | 8 | | | X | X | | DIG | 7 + 2 | X | | | | (≙ Mini-SR8) |
| UPD-10 | AL | | | 28 | X | | 1 | 10 | X | X | | | | DIG | 10 + 1 | X | | X | Drucker | sehr kleiner Tischrechner |
| ● **BOHSEI** | | | | | | | | | | | | | | | | | | | | |
| 710 | ARL | | | 30 | X | | 1 | 12 | X | X | | | X | | | | | X | Drucker (12 Stellen) | kl. Tischrechner/%/M+/M– |
| 720 | ARL | | | 30 | X | | 1 | 12 | X | X | | | X | DIG | 12 + 1 | | | X | wie 710 | wie 710 |
| 810 | ARL | | | 30 | X | | 1 | 12 | X | X | | | X | | | | | X | wie 710 | wie 710 |
| 820 | ARL | | | 30 | X | | 1 | 12 | X | X | | | X | DIG | 12 + 1 | | | X | wie 710 | wie 710 |
| 850 | ARL | | | 30 | X | | 1 | 12 | X | X | | | X | DIG | 12 + 1 | | | X | wie 710 | wie 710 |
| LC-86 | AL | | | 21 | X | | 1 | 8 | | X | | | | LCD | 8 + 1 | | X | | | %/M+/M– |
| ● **BRAUN** | | | | | | | | | | | | | | | | | | | | |
| ET33 | AL | | 1 | 25 | X | | 1 | 8 | X | X | | | | LCD | 8 + 1 | | X | | | %/△ %/+/– /M+/M– |
| ET44 | AL | | 1 | 25 | X | | 1 | 8 | | X | | | | LCD | 8 + 1 | | X | | | %/+/– /M+/M– |
| ● **Brother** | | | | | | | | | | | | | | | | | | | | |
| 110L | AL | | 1 | 24 | X | | 1 | 9 | X | X | | | X | LCD | 9 + 1 | | X | | | %/+/– /M+/M–/1500 Stunden |
| 122 PDR | ARL | | 1 | 31 | X | | 1 | 12 | X | X | | | X | DIG | 12 + 1 | | | X | Drucker | Tischrechner %/M+/M– |
| 408AD | AL | | 1 | 22 | X | | | | | X | | | | DIG | 8 + 1 | X | | X | | π/%/+/– |
| 408AZ | AL | | 1 | 22 | X | | | | | X | | | | DIG | 8 + 1 | | X | | | π/%/+ /– |
| 418E | AL | | 3 | 24 | X | | | | | X | | | | DIG | 8 + 1 | | X | | | π/%/+/– |
| 418ER | AL | | 3 | 24 | X | | | | | X | | | | DIG | 8 + 1 | X | | X | | π/%/+/– |
| 508AD | AL | | 1 | 25 | X | | 1 | 8 | | X | | | | DIG | 8 + 1 | X | | X | | π/%/+/– /M+/M– |
| 508AY | AL | | 1 | 22 | X | | 1 | 8 | | X | | | | DIG | 8 + 1 | | X | X | | %/+/– |
| 508E | AL | | 1 | 25 | X | | 1 | 8 | | X | | | | DIG | 8 + 1 | | X | | | π/%/+/– /M+/M– |
| 508ER | AL | | 1 | 25 | X | | 1 | 8 | | X | | | | DIG | 8 + 1 | X | | X | | π/%/+/– /M+/M– |
| 508S | AL | | 12 | 25 | | X | 1 | 8 | | X | | X | | DIG | 8 + 1 | X | | X | | π/rad |
| 512SR | AL | | 20 | 40 | X | | 1 | 10 | | | X | X | | DIG | 10 + 2 | X | | X | | π/+/–/rad/vier Löschtasten |
| 518E | AL | | 3 | 28 | X | | 1 | 8 | | X | | | | DIG | 8 + 1 | | X | | | π/%/+/– /M+/M– |
| 518ER | AL | | 3 | 28 | X | | 1 | 8 | | X | | | | DIG | 8 + 1 | X | | X | | π/%/+/– /M+/M– |
| 612P | ARL | | | 25 | X | | 1 | 6 | X | X | | | X | | | | | X | Drucker (6 Stellen) | Tischrechner/%/M+ |
| 618ACL | AL | | 1 | 24 | | X | 1 | 8 | | X | | | | LCD | 8 + 1 | | X | | | %/M+/M–/Kalender/Stoppuhr/Wecker/ 10 000 Stunden |
| 712SR | AL | 2 | 19 | 45 | X | | 1 | 8 | | | X | X | | DIG | 10 + 2 | | X | | | π/+/– /rad/hyp. |
| 722SR | AL | 2 | 19 | 45 | X | | 1 | 8 | | | X | X | | DIG | 10 + 2 | X | | X | | π/+/– /rad/hyp. |
| 715SR | AL | 2 | 22 | 46 | X | | 1 | 8 | | | X | X | | DIG | 10 + 2 | | X | | | π/+/– /rad/hyp. |
| 718SR | AL | 3 | 17 | 41 | | X | 1 | 8 | | | X | X | | LCD | 8 + 1 | | X | | | π/+/– /%/M+/Statistik |
| 725SR | AL | 2 | 22 | 46 | X | | 1 | 8 | | | X | X | | DIG | 10 + 2 | X | | X | | π/+/– /rad/hyp./Statistik |
| 738SR | AL | 2 | 22 | 40 | | X | 1 | 8 | | | X | X | | LCD | 10 + 2 | | X | | | π/△ %/+/– /M+/Statistik/1500 Stunden |
| 818L | AL | | 1 | 24 | X | | 1 | 8 | | X | | | | LCD | 8 + 1 | | X | | | %/+/– /M+/M–/2 000 Stunden |
| 828L | AL | | 1 | 24 | X | | 1 | 8 | | X | | | | LCD | 8 + 1 | | X | | | %/+/– /M+/M–/3 000 Stunden |
| 838SL | AL | | 1 | 26 | X | | 1 | 8 | | X | | | | LCD | 8 + 1 | | X | | | %/△ %/+/– /M+/M–/1 600 Stunden |
| LD12 | AL | | | 37 | | | | | | X | | | X | LCD | 12 + 2 | | X | | | %/△ % |
| ● **Brunsviga** | | | | | | | | | | | | | | | | | | | | |
| 1200M | ARL | | | 28 | X | | 1 | 12 | X | X | | | X | | | X | | X | Drucker (12 Stellen) | %/Tischrechner |
| 1200K | ARL | | | 22 | X | | | | X | X | | | X | | | | | X | Drucker (12 Stellen) | %/Tischrechner |
| 1210M | ARL | | | 28 | X | | 1 | 12 | X | X | | | X | | | X | | X | Drucker (12 Stellen) | %/Tischrechner |
| 1410P | ARL | | | 28 | X | | 1 | 12 | X | X | | | X | | | | | X | Drucker (12 Stellen) | %/Tischrechner |
| 1620P | ARL | | 1 | 33 | X | | 2 | 12 | X | X | | | X | | | | | X | Drucker (12 Stellen) | %/Tischrechner |

| Hersteller Typ | Rechenlogik | Anzahl der Klammern | Anzahl der Funktionen | Anzahl der Tasten | Tastenbeleg. einfach | Tastenbeleg. mehrfach | Zahl der Speicher | Zahl der Stellen | Komma Fest- | Komma Fließ- | Komma Gleit- | techn.-wiss. | kaufmännisch | Anzeige Art | Anzeige Stellen | Stromversorgung Akku | Stromversorgung Batterie | Stromversorgung Netz | Ergänzungen | Besonderheiten |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| noch | | | | | | | | | | | | | | | | | | | | |
| • **Brunsviga** | | | | | | | | | | | | | | | | | | | | |
| 1810P | ARL | | | 30 | X | | 1 | 12 | X | X | | | X | | | | | X | Drucker (16 Stellen) | %/Tischrechner |
| 4020D | ARL | | | 26 | X | | 1 | 12 | X | X | | | X | DIG | 12 | | | X | | %/Tischrechner |
| 5020D | ARL | | | 26 | X | | 2 | 14 | X | X | | | X | DIG | 14 | | | X | | %/Tischrechner |
| 6030D | AL | | 19 | 35 | X | | 1 | 10 | | | X | X | | DIG | 12 + 2 | | | X | | π/+/_ /rad/M+/M–/Tischrechner |
| Der Braunschweiger | AL | | 12 | 20 | | X | 1 | 8 | | | X | X | | DIG | 8 + 1 | | X | | | π/M+/M–/rad/Querformat |
| • **Canon** | | | | | | | | | | | | | | | | | | | | |
| 8-T | AL | | | 24 | X | | 1 | 8 | X | X | | | | DIG | 8 + 1 | | X | X | | Uhr/M+/+/_ |
| F-5 | AL | | 12 | 26 | | X | 1 | 8 | | X | | X | | LED | 8 + 1 | X | X | X | | π/M+/rad/+/_ |
| F-7 | AL | 7 | 20 | 36 | | X | 2 | 10 | | | X | X | | LED | 10 + 2 | X | X | X | | π/M+/rad/+/_ /Bruchstrich/Umrechnungen |
| F-10 | AL | | 13 | 40 | X | | 2 | 12 | X | X | | X | | LED | 12 + 1 | | | X | | Tischrechner/M+/+/_ /rad/Polar |
| F-10P | AL | | 13 | 40 | X | | 2 | 12 | X | X | | X | | LED | 12 + 1 | | | X | Drucker (12 Stellen) | wie F-10 |
| F-11 | AL | 3 | 36 | 48 | X | | 10 | 12 | X | | X | X | | LED | 12 + 3 | | | X | | Tischrechner/+/_ /π/M+/rad/Polar/Statistik/Umrechnungen |
| F-20P | AL | | 18 | 41 | X | | 1 | 12 | X | X | | X | X | | | | | | Drucker (12 Stellen) | Tischrechner für Statistik und Finanzen |
| F-31 | AL | 1 | 3 | 30 | X | | 1 | 8 | X | X | | | | DIG | 8 + 1 | | X | X | | π/+/_/M+ |
| F-41 | AL | | 15 | 30 | X | | 1 | 8 | | | X | | | DIG | 7 + 2 | | X | X | | π/+/_ /M+/rad |
| F-52 | AL | 2 | 22 | 35 | | X | 1 | 8 | | | X | X | | LCD | 7 + 2 | | X | | | π/+/_/M+/M–/Hyp./Statistik |
| F-61 | AL | 1 | 21 | 40 | | X | 1 | 8 | | | X | X | | DIG | 10 + 2 | X | | X | | π/+/_ /M+/rad/Statistik |
| F-62 | AL | 2 | 30 | 37 | | X | 1 | 10 | | | X | X | | LCD | 10 + 2 | | X | | | wie F-52 |
| F-71 | AL | 1 | 30 | 43 | X | | 5 | 10 | | | X | X | | DIG | 12 + 2 | X | | X | | π/M+/+/_ /rad/Polar/Statistik |
| L100S | ARL | | | 18 | X | | | | X | X | | | X | LED | 10 + 1 | | | X | | Tischrechner |
| L121F | ARL | | | 26 | X | | 1 | 12 | X | X | | | X | LED | 12 + 1 | | | X | | Tischrechner/%/M+ |
| L800 | ARL | | | 17 | X | | | | | X | | | X | LED | 8 + 1 | | | X | | Tischrechner |
| L804 | AL | | | 19 | X | | | | | X | | | | DIG | 8 + 1 | | X | X | | Tischrechner/% |
| L810 | AL | | | 19 | X | | 1 | 8 | X | X | | | X | LED | 8 + 1 | | X | X | | Tischrechner |
| L812 | AL | | | 22 | X | | 1 | 8 | X | X | | | X | DIG | 8 + 1 | | X | X | | Tischrechner/%/M+ |
| L813 | AL | | | 22 | X | | 1 | 8 | X | X | | | X | DIG | 8 + 1 | | X | X | | Tischrechner/%/M+ |
| L1000 | ARL | | | 19 | X | | | | X | X | | | X | LED | 10 + 1 | | | X | | Tischrechner/% |
| L1010 | ARL | | | 22 | X | | 1 | 10 | X | X | | | X | DIG | 10 + 1 | | X | X | | Tischrechner/M+/M–/% |
| L1210 | ARL | | 1 | 27 | X | | 1 | 12 | X | X | | | X | LED | 12 + 1 | | | X | | Tischrechner/M+/% |
| L1212 | ARL | | | 22 | X | | 1 | 12 | X | X | | | X | LED | 12 + 1 | | | X | | Tischrechner/M+/% |
| L1214 | ARL | | 1 | 27 | X | | 1 | 12 | X | X | | | X | DIG | 12 + 1 | | X | X | | Tischrechner/M+/% |
| L1217 | ARL | | 1 | 28 | X | | 1 | 12 | X | X | | | X | DIG | 12 + 1 | | | X | | Tischrechner/%/M+ |
| L1420 | ARL | | | 33 | X | | 2 | 14 | X | X | | | X | LED | 14 + 1 | | | X | | Tischrechner/%/M+ |
| L1422 | ARL | | 1 | 29 | X | | 2 | 14 | X | X | | | X | LED | 14 + 1 | | | X | | Tischrechner/%/M+ |
| L1423 | ARL | | 1 | 32 | X | | 2 | 14 | X | X | | | X | DIG | 14 + 1 | | | X | | Tischrechner/%/M+ |
| L1630 | ARL | | 1 | 38 | X | | 3 | 16 | X | X | | | X | LED | 16 + 1 | | | X | | Tischrechner/%/M+ |
| L1632 | ARL | | 1 | 33 | X | | 3 | 16 | X | X | | | X | LED | 16 + 1 | | | X | | Tischrechner/%/M+ |
| LC-4 | AL | | 1 | 24 | X | | 1 | 8 | | X | | | | LCD | 8 + 1 | | X | | | %/M+/M–/+/_ |
| LC-5 | AL | | 1 | 24 | X | | 1 | 8 | | X | | | | LCD | 8 + 1 | | X | | | %/M+/M–/+/_ |
| LC-6 | AL | | 1 | 24 | X | | 1 | 8 | | X | | | | LCD | 8 + 1 | | X | | | %/+/_ /M+/M–/3 000 Stunden |
| LC-8M | AL | | 1 | 24 | X | | 1 | 8 | | X | | | | LCD | 8 + 1 | | X | | | %/M+/M–/+/_ |
| LC-61T | AL | | 1 | 24 | | X | 1 | 8 | | X | | | | LCD | 8 + 1 | | X | | | wie LC-6/Uhr/Stopp/Kalender/Wecker/1 Jahr |
| LC-815T | AL | | 1 | 28 | | X | 4 | 8 | | X | | | | LCD | 8 + 1 | | X | | | %/M+/M–/wie LC-61T |
| LC-1014 | AL | | 1 | 25 | X | | 1 | 10 | X | X | | | | LCD | 10 + 1 | | X | | | %/M+/M–/+/_ |
| LC-Quartz | AL | | 1 | 29 | X | | 1 | 8 | | X | | | | LCD | 8 + 1 | | X | | | %/M+/M–/+/_ /Quarzuhr |
| LD-8S | AL | | 1 | 20 | X | | | | | X | | | | LED | 8 + 1 | X | | X | | % |
| LD-8MS | AL | | 1 | 24 | X | | 1 | 8 | | X | | | | LED | 8 + 1 | X | | X | | %/M+/M– |
| LD-10M3 | AL | | 1 | 28 | X | | 1 | 10 | X | X | | | | LED | 10 + 1 | | X | X | | %/M+/M–/+/_ |
| LD-80 | AL | | | 24 | X | | | | | X | | | | DIG | 8 + 1 | | X | X | | % |
| LD-Multi 8 | AL | | 1 | 24 | X | | 1 | 8 | | X | | | | DIG | 8 + 1 | | X | X | | +/_ /M+/2 Leuchtanzeigen |
| LD-Ruler 8 | AL | | | 22 | X | | 1 | 8 | | X | | | | LCD | 8 + 1 | | X | | | %/M+/M–/Querformat |
| LE-84 | AL | | | 18 | X | | | | | X | | | | LED | 8 + 1 | | X | X | | |
| LE-85 | AL | | 1 | 20 | X | | | | | X | | | | LED | 8 + 1 | | X | X | | % |
| LE-100 | AL | | | 18 | X | | | | | X | | | | LED | 10 + 1 | X | X | X | | |

| Hersteller Typ | Rechenlogik | Anzahl der Klammern | Anzahl der Funktionen | Anzahl der Tasten | Tastenbeleg. einfach | Tastenbeleg. mehrfach | Zahl der Speicher | Zahl der Stellen | Komma Fest- | Komma Fließ- | Komma Gleit- | techn.-wiss. | kaufmännisch | Anzeige Art | Anzeige Stellen | Stromversorgung Akku | Stromversorgung Batterie | Stromversorgung Netz | Ergänzungen | Besonderheiten |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| noch | | | | | | | | | | | | | | | | | | | | |
| • **Canon** | | | | | | | | | | | | | | | | | | | | |
| MP142R | ARL | | 1 | 36 | X | | 2 | 14 | X | X | X | | X | | | | | X | Drucker (14 Stellen) | Tischrechner/%/M+/M– |
| MP1210 | ARL | | | 27 | X | | 1 | 12 | X | X | | | X | | | | | X | Drucker (12 Stellen) | Tischrechner/%/M+ |
| MP1212 | ARL | | | 27 | X | | 1 | 12 | X | X | | | X | LED | 12 + 1 | | | X | Drucker | Tischrechner/%/M+ |
| MP1217 | ARL | | | 28 | X | | 1 | 12 | X | X | | | X | | | | | X | Drucker (12 Stellen) | Tischrechner/%/M+ |
| MP1422 | ARL | | 1 | 36 | X | | 2 | 14 | X | X | | | X | | | | | X | Drucker (14 Stellen) | Tischrechner/%/M+/M– |
| P10-D | AL | | | 28 | X | | 1 | 10 | X | X | | | | LED | 10 + 1 | X | | X | Drucker | %/M+/M– |
| P21-D | AL | | | 32 | X | | 1 | 10 | X | X | | | | LED | 10 + 1 | | | X | Drucker | Tischrechner/%/M+/M– |
| P1010 | ARL | | | 21 | X | | 1 | 10 | X | X | | | X | | | | | X | Drucker (10 Stellen) | Tischrechner/%/M+ |
| P1011-D | ARL | | | 25 | X | | 1 | 10 | X | X | | | X | LED | 10 + 1 | | | X | Drucker | Tischrechner/%/M+ |
| P1210 | ARL | | | 24 | X | | 1 | 12 | | X | | | X | | | | | X | Drucker (12 Stellen) | Tischrechner/%/M+ |
| P1211-D | ARL | | | 29 | X | | 1 | 12 | X | X | | | X | LED | 12 + 1 | | | X | Drucker | Tischrechner/%/M+ |
| P1217 | ARL | | | 28 | X | | 1 | 12 | X | X | | | X | | | | | X | Drucker (12 Stellen) | Tischrechner/%/M+ |
| P1014 | ARL | | | 25 | X | | 1 | 10 | X | X | | | X | | | | | X | Drucker (10 Stellen) | Tischrechner/% |
| P1014-D | ARL | | | 25 | X | | 1 | 10 | X | X | | | X | LED | 10 + 1 | | | X | wie P1014 | wie P1014 |
| P1212-D | ARL | | | 32 | X | | 1 | 12 | X | X | | | X | LED | 12 + 1 | | | X | Drucker | wie P1014 |
| P1218-D | ARL | | | 28 | X | | 1 | 12 | X | X | | | X | DIG | 12 + 1 | | | X | Drucker | Tischrechner/%/M+ |
| P1250 | ARL | | | 28 | X | | 1 | 12 | X | X | | | X | | | | | X | Drucker (12 Stellen) | wie P1014 |
| P1250-D | ARL | | | 28 | X | | 1 | 12 | X | X | | | X | LED | 12 + 1 | | | X | wie P1250 | wie P1250 |
| P1410 | ARL | | | 30 | X | | 1 | 14 | X | X | | | | | | | | X | Drucker (14 Stellen) | Tischrechner/%/M+ |
| P1410-D | ARL | | | 30 | X | | 1 | 14 | X | X | | | X | LED | 14 + 1 | | | X | Drucker | Tischrechner/%/M+ |
| P1420-D | ARL | | | 34 | X | | 2 | 14 | X | X | | | X | LED | 14 + 1 | | | X | Drucker | Tischrechner/%/M+ |
| Pocketronic II | AL | | | 19 | X | | | | | X | | | | | | X | | X | Drucker (10 Stellen) | % |
| • **Casio** | | | | | | | | | | | | | | | | | | | | |
| 102-MR | AL | | 1 | 23 | X | | 1 | 10 | | X | | | | DIG | 10 + 1 | | X | X | | %/M+/M– |
| 121-MR | AL | | 1 | 26 | X | | 1 | 12 | X | X | | | | DIG | 12 + 1 | | X | X | | %/M+/M–/+/– |
| 802-MR | AL | | 1 | 23 | X | | 1 | 8 | X | X | | | | DIG | 8 + 1 | | X | X | | %/M+/M– |
| AL-8 | AL | | 5 | 24 | | X | 2 | 8 | X | X | | X | X | DIG | 8 + 1 | | X | X | | %/M+/M–/Statistik/Zeitrechnung |
| AL-10 | AL | | 5 | 24 | | X | 2 | 10 | X | X | | X | X | DIG | 10 + 1 | | X | X | | wie AL-8 |
| AL-10S | AL | | 5 | 32 | X | | 2 | 10 | X | X | | X | X | DIG | 10 + 1 | | X | X | | %/Δ %/200-jähriger Kalender |
| AQ-1000 | AL | | 1 | 28 | X | | 1 | 8 | | X | | | | LCD | 8 + 1 | | X | | | %/M+/M–/Stoppuhr/Wecker/10 000 Stunden |
| B-1 | AL | | 1 | 23 | X | | 1 | 8 | | X | | | | DIG | 8 + 1 | X | | X | | %/Δ %/M+/M– |
| Biolator | AL | | | 20 | X | | | | | X | | | | DIG | 8 + 1 | | | X | | Datum, Biorhythmus |
| CQ-1 | AL | | | 23 | X | | | | | | | | | DIG | 8 + 1 | | X | X | | Datum, Stoppuhr /Wecker |
| CQ-81 | AL | | 1 | 28 | X | | 1 | 8 | | X | | | | LCD | 8 + 1 | | X | | | %/M+/M–/Datum/Uhr/Wecker/13 000 Stunden |
| CQ-82 | AL | | 1 | 26 | X | | | | | X | | | | LCD | 8 + 1 | | X | | | %/Δ %/Datum/Wecker/Uhr |
| DR-1210 | ARL | | | 32 | X | | 1 | 12 | X | X | | | X | DIG | 12 + 1 | | | X | Drucker | Tischrechner/%/Δ %/M+/M– |
| DR-1211 | ARL | | | 32 | X | | 1 | 12 | X | X | | | X | DIG | 12 + 1 | | | X | Drucker | Tischrechner/%/Δ %/+/– M+/M– |
| DR-1410 | ARL | | | 32 | X | | 1 | 14 | X | X | | | X | DIG | 14 + 1 | | | X | Drucker | wie DR-1210 |
| F-1 | ARL | | | 29 | X | | 1 | 10 | X | X | | | X | DIG | 10 + 1 | | X | X | | Tischrechner/%/Δ %/M+/M– |
| F-2 | ARL | | | 29 | X | | 1 | 12 | X | X | | | X | DIG | 12 + 1 | | X | X | | Tischrechner/%/Δ %/M+/M– |
| F-3 | ARL | | | 29 | X | | 2 | 16 | X | X | | | X | DIG | 16 + 1 | | X | X | | Tischrechner/%/Δ %/M+/M– |
| FL-1 | ARL | | 1 | 29 | X | | 1 | 10 | X | X | | | X | LCD | 10 + 1 | | X | | | kleiner Tischrechner/%/Δ %/+/– /M+/M–/ 10 000 Stunden |
| FL-2 | ARL | | 1 | 29 | X | | 1 | 12 | X | X | | | X | LCD | 12 + 1 | | X | | | wie FL-1 |
| fx-10 | AL | | 10 | 29 | X | | | | | X | | X | | DIG | 8 + 1 | | X | X | | π |
| fx-15 | AL | | 13 | 37 | X | | 1 | 8 | | | X | X | | DIG | 8 + 2 | | X | X | | π/M+/M–/+/– |
| fx-17 | AL | | 13 | 37 | X | | 1 | 8 | | | X | X | | DIG | 8 + 2 | | X | X | | π/M+/M–/+/– |

| Hersteller Typ | Rechenlogik | Anzahl der Klammern | Anzahl der Funktionen | Anzahl der Tasten | Tastenbeleg. einfach | Tastenbeleg. mehrfach | Zahl der Speicher | Zahl der Stellen | Komma Fest- | Komma Fließ- | Komma Gleit- | techn.-wiss. | kaufmännisch | Anzeige Art | Anzeige Stellen | Stromversorgung Akku | Stromversorgung Batterie | Stromversorgung Netz | Ergänzungen | Besonderheiten |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| noch | | | | | | | | | | | | | | | | | | | | |
| • **Casio** | | | | | | | | | | | | | | | | | | | | |
| fx-19 | AL | | 20 | 38 | X | | 1 | 8 | | | X | X | | DIG | 8 + 2 | | X | X | | $\pi$/M+/M–/$^+$/_ /rad/Hexadezimal-Umrechnung/Statistik/Bruchrechnung |
| fx-20 | AL | | 11 | 32 | X | | 1 | 8 | | | X | X | | DIG | 8 + 2 | | X | X | | M+/$^+$/_ /Hexadezimal-Umrechnung |
| fx-29 | AL | | 15 | 38 | X | | 1 | 8 | | | X | X | | DIG | 8 + 2 | X | | X | | $\pi$/M+/M–/$^+$/_ /Hexadezimal-Umrechnung/Bruchrechnung/Statistik |
| fx-31 | AL | 6 | 20 | 35 | | X | 1 | 8 | | | X | X | | DIG | 8 + 2 | | X | | | $\pi$/$^+$/_ /M+/M–/Statistik/Umrechnungen |
| fx-39 | AOS | 6 | 26 | 38 | X | | 1 | 8 | | | X | X | | DIG | 8 + 2 | X | X | X | | $\pi$/M+/M–/$^+$/_/Statistik/Bruchrechnung/Hexadezimal-Umrechnung |
| fx-48 | AL | 2 | 20 | 22 | | X | 1 | 8 | | X | | X | | LCD | 8 + 1 | | X | | | $\pi$/M+/M–/Statistik |
| fx-58 | AL | 1 | 24 | 20 | | X | 1 | 8 | | X | X | X | | LCD | 8 + 2 | | X | | | $\pi$/$^+$/_ /M+/M–/Uhr/Wecker |
| fx-80 | AL | 6 | 24 | 38 | | X | 1 | 8 | | | X | X | | LCD | 8 + 2 | | X | | | $\pi$/$^+$/_ /M+/M–/rad/Statistik |
| fx-101 | AL | | 13 | 37 | X | | 1 | 8 | | | X | X | | DIG | 10 + 2 | | X | X | | $\pi$/M+/M–/$^+$/_ /Hexadezimal-Umrechnung |
| fx-102 | AL | | 20 | 38 | X | | 1 | 8 | | | X | X | | DIG | 10 + 2 | | X | X | | wie fx-19 |
| fx-105 | AL | | 21 | 35 | X | | 1 | 8 | | | X | X | | DIG | 10 + 2 | X | | X | | $\pi$/M+/M–/$^+$/_ /rad/Bruchrechnung/Hexadezimal-Umrechnung |
| fx-110 | AL | | 23 | 35 | | X | 1 | 8 | | | X | X | | DIG | 10 + 2 | X | | X | | wie fx-105, zusätzlich: Statistik/dB-Berechnung |
| fx-120 | AOS | 6 | 27 | 39 | | X | 1 | 8 | | | X | X | | DIG | 10 + 2 | X | X | X | | $\pi$/M+/M–/$^+$/_ /%/rad/Statistik/Hexadezimal-Umwandlung/Bruchrechnung |
| fx-140 | AL | | 26 | 38 | | X | 1 | 12 | | | X | X | | DIG | 10 + 2 | | X | | | wie fx-31 |
| fx-1000 | AL | | 23 | 37 | | X | 1 | 8 | | | X | X | | LCD | 8 + 2 | | X | | | $\pi$/M+/M–/$^+$/_ /rad/Statistik/Bruchrechnung/Hexadezimal-Umrechnung |
| fx-2000 | AL | | | 37 | | X | 1 | 8 | | | X | X | | LCD | 8 + 2 | | X | | | wie fx-1000/1000 Stunden |
| fx-3000 | AL | 2 | 27 | 39 | | X | 1 | 10 | | | X | X | | LCD | 10 + 2 | | X | | | wie fx-120 |
| fx-3100 | AL | 2 | 30 | 38 | | X | 1 | 10 | | | X | X | | LCD | 10 + 2 | | X | | | $\pi$/%/$^+$/_ /M+/M–/rad/Statistik |
| fx-5000 | AL | 5 | 36 | 38 | | X | 10 | 12 | | | X | X | | LCD | 12 + 2 | | X | | | $\pi$/%/$^+$/_ /M+/M–/Statistik |
| fx-5000EP | AL | | 15 | 47 | X | | 7 | 10 | | | X | X | | DIG | 12 + 2 | | | X | Drucker | Tischrechner/$\pi$/M+/M–/$^+$/_ /rad/Hexadezimal-Umrechnung |
| fx-8000 | AL | 2 | 30 | 38 | | X | 1 | 8 | | X | | X | | LCD | 8 + 1 | | X | | | $\pi$/$^+$/_ /M+/M–/Statistic/Uhr/1300 Stunden |
| HL-101 | AL | | 1 | 25 | X | | 1 | 10 | X | X | | | | LCD | 10 + 1 | | X | | | %/$^+$/_ /M+/M–/7000 Stunden |
| HL-801 | AL | | 1 | 23 | X | | 1 | 8 | | X | | | | LCD | 8 + 1 | | X | | | wie HL-101 |
| IL-20 | AL | | 1 | 24 | X | | 1 | 8 | X | X | X | | | LCD | 10 + 2 | | X | | | %/M+/M–/2500 Stunden |
| JR-101 | ARL | | 1 | 29 | X | | 1 | 9 | X | X | | | X | | | | X | X | Drucker (9 Stellen) | kleiner Tischrechner/%/$^+$/_ /M+/M– |
| JR-110 | ARL | | 1 | 29 | X | | 1 | 9 | X | X | | | X | DIG | 9 + 1 | | X | X | wie JR-101 | wie JR-101 |
| LC-78 | AL | | | 22 | X | | 1 | 8 | | X | | | | LCD | 8 + 1 | | X | | | %/M+/M–/$\Delta$%/800 Stunden |
| LC-78G | AL | | | 22 | X | | 1 | 8 | | X | | | | LCD | 8 + 1 | | X | | | %/M+/M– |
| LC-78S | AL | | 1 | 22 | X | | 1 | 8 | | X | | | | LCD | 8 + 1 | | X | | | % |
| LC-79 | AL | | 1 | 23 | X | | 1 | 8 | | X | | | | LCD | 8 + 1 | | X | | | %/M+/M– |
| LC-81 | AL | | | 20 | X | | 1 | 8 | | X | | | | LCD | 8 + 1 | | X | | | % |
| LC-88 | AL | | 1 | 24 | X | | 1 | 8 | | X | | | | LCD | 8 + 1 | | X | | | %/M+/M–/2000 Stunden |
| LC-820 | AL | | 1 | 23 | X | | 1 | 8 | | X | | | | LCD | 8 + 1 | | X | | | %/$\Delta$%/M+/M–/1000 Stunden |
| LC-821 | AL | | 1 | 23 | X | | 1 | 8 | | X | | | | LCD | 8 + 1 | | X | | | %/$\Delta$%/M+/M–/2000 Stunden |
| LC-822 | AL | | 1 | 23 | X | | 1 | 8 | | X | | | | LCD | 8 + 1 | | X | | | %/$\Delta$%/M+/M–/2500 Stunden |
| LC-825 | AL | | 1 | 23 | X | | 1 | 8 | | X | | | | LCD | 8 + 1 | | X | | | %/M+/M–/2500 Stunden |
| LC-826 | AL | | 1 | 23 | X | | 1 | 8 | | X | | | | LCD | 8 + 1 | | X | | | %/$\Delta$%/M+/M–/4000 Stunden |
| LC-841 | AL | | 1 | 23 | X | | 1 | 8 | | X | | | | LCD | 8 + 1 | | X | | | wie LC-826 |
| LC-1010 | AL | | 1 | 23 | X | | 1 | 10 | X | X | | | | LCD | 10 + 1 | | X | | | wie LC-822 |
| LC-1021 | AL | | 1 | 25 | X | | 1 | 10 | X | X | | | | LCD | 10 + 1 | | X | | | %/$^+$/_ /M+/M– |
| M-1 | AL | | 1 | 21 | X | | 1 | 8 | | X | | | | DIG | 8 + 1 | | X | | | %/$\Delta$% |
| M-810 | AL | | | 20 | X | | 1 | 8 | | X | | | | LCD | 8 + 1 | | X | | | % |
| MC-811 | AL | | | 20 | X | | 1 | 8 | | X | | | | LCD | 8 + 1 | | X | | | % |
| MQ-1 | AL | | | 23 | X | | | | | X | | | | LCD | 8 + 1 | | X | | | Quarzuhr/Stoppuhr/13000 Stunden |
| MQ-10 | AL | | | 24 | X | | | | | X | | | | LCD | 8 + 1 | | X | | | %/Datum/Uhr/Wecker/13000 Stunden |
| MQ-11 | AL | | 1 | 29 | X | | | | | X | | | | LCD | 8 + 1 | | X | | | %/Uhr/Datum/200 Jahre Kalender/Biorhythmus |
| Melody-80 | AL | | 1 | 30 | | X | 1 | 8 | | X | | | | LCD | 8 + 1 | | X | | | %/$\Delta$%/Stoppuhr/Wecker/Musik |
| Mem.-8A | AL | | 1 | 23 | X | | 1 | 8 | X | X | | | | DIG | 8 + 1 | | X | X | | %/M+/M– |
| Mem.-8F | AL | | 1 | 23 | X | | 1 | 8 | X | X | | | | DIG | 8 + 1 | | X | X | | %/M+/M– |
| Mem.-8R | AL | | | 20 | X | | 1 | 8 | | X | | | | DIG | 8 + 1 | | X | X | | %/M+ |

| Hersteller Typ | Rechenlogik | Anzahl der Klammern | Anzahl der Funktionen | Anzahl der Tasten | Tastenbeleg. einfach | Tastenbeleg. mehrfach | Zahl der Speicher | Zahl der Stellen | Komma Fest- | Komma Fließ- | Komma Gleit- | techn.-wiss. | kaufmännisch | Anzeige Art | Anzeige Stellen | Stromversorgung Akku | Stromversorgung Batterie | Stromversorgung Netz | Ergänzungen | Besonderheiten |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| noch | | | | | | | | | | | | | | | | | | | | |
| ● **Casio** | | | | | | | | | | | | | | | | | | | | |
| Mem.-8S | AL | | 1 | 21 | X | | 1 | 8 | | X | | | | DIG | 8 + 1 | | X | X | | %/M+ |
| Mem.-10 | AL | | 1 | 23 | X | | 1 | 8 | X | X | | | | DIG | 10 + 1 | | X | X | | %/M+/M– |
| Micro-Mini | AL | | | 20 | X | | 1 | 8 | | X | | | | LCD | 8 + 1 | | X | | | %/M+ |
| Mini-8 | AL | | | 20 | X | | | | | X | | | | DIG | 8 + 1 | | X | X | | % |
| Mini-Printer | AL | | 1 | 23 | X | | 1 | 8 | | X | | | | DIG | 8 + 1 | X | | X | Drucker | |
| Personal-I | AL | | 1 | 20 | X | | | | | X | | | | DIG | 8 + 1 | | X | X | | % |
| Personal-8 | AL | | | 19 | X | | | | | X | | | | DIG | 8 + 1 | | X | X | | % |
| Personal-8M | AL | | | 23 | X | | 1 | 8 | | X | | | | DIG | 8 + 1 | | X | X | | %/M+/M–/+/– |
| Personal M-1 | AL | | 1 | 25 | X | | 1 | 8 | | X | | | | DIG | 8 + 1 | | X | X | | % |
| Personal-Mini | AL | | | 19 | X | | | | | X | | | | DIG | 6 + 6 + 1 | | X | X | | Querformat/Doppelanzeige: 6 + 6 Stellen |
| Pocket-8S | AL | | | 19 | X | | | | | X | | | | DIG | 8 + 1 | | X | X | | % |
| Pocket-LC | AL | | | 21 | X | | 1 | 8 | | X | | | | LCD | 8 + 1 | | X | | | % |
| Pocket-Mini | AL | | 1 | 20 | X | | | | | X | | | | DIG | 8 + 1 | | | | | % |
| QL-10 | AL | | | 24 | X | | | | | X | | | | LCD | 8 + 1 | | X | | | %/Kalender/Uhr/Wecker/Feuerzeug |
| R-110 | ARL | | | 25 | X | | 2 | 10 | X | X | | | X | DIG | 10 + 1 | | | X | Drucker | Tischrechner/%/M+/M– |
| R-120 | ARL | | | 25 | X | | 3 | 10 | X | X | | | X | DIG | 10 + 1 | | | X | Drucker | Tischrechner/%/M+/M– |
| R-200 | ARL | | 1 | 28 | X | | 3 | 12 | X | X | | | X | | | | | X | Drucker (12 Stellen) | Tischrechner/%/M+/M– |
| R-220 | ARL | | | 30 | X | | 2 | 12 | X | X | | | X | DIG | 12 + 1 | | | X | Drucker | Tischrechner/%/M+/M– |
| R-320 | ARL | | 1 | 31 | X | | 3 | 16 | X | X | | | X | DIG | 16 + 1 | | | X | Drucker | Tischrechner/%/M+/M– (2 mal) |
| R-1201 | ARL | | | 28 | X | | 1 | 12 | X | X | | | X | | | | | X | Drucker (12 Stellen) | Tischrechner/%/M+/M– |
| R-1203 | ARL | | | 29 | X | | 1 | 12 | X | X | | | X | | | | | | Drucker (12 Stellen) | Tischrechner/%/+/– |
| ST-1 | AL | | 1 | 25 | X | | 1 | 8 | | X | | | | DIG | 8 + 1 | | X | X | | %/M+/M–/Stoppuhr |
| ST-24 | AL | | | 22 | | X | 1 | 8 | | X | | | | LCD | 8 + 1 | | X | | | %/M+/M–/Stoppuhr/Termine |
| ● **CBM** | | | | | | | | | | | | | | | | | | | | |
| cbm | AL | | 13 | 19 | | X | 1 | 8 | | | X | X | | DIG | 7 + 2 | | X | | | π/e/+/–/M+ |
| SR36 | AL | 2 | 12 | 36 | X | | 1 | 10 | | | X | X | | DIG | 12 + 2 | | X | | | π/+/–/rad |
| ● **CI** | | | | | | | | | | | | | | | | | | | | |
| 50P | ARL | | | 23 | X | | | | X | X | | | X | | | | | X | Drucker (12 Stellen) | Tischrechner/% |
| 65S | ARL | | | 25 | X | | 1 | 12 | X | X | | | X | | | | | X | Drucker (12 Stellen) | Tischrechner/%/M+ |
| 100S | ARL | | | 27 | X | | 1 | 12 | X | X | | | X | | | | | X | Drucker (12 Stellen) | Tischrechner/%/M+/M– |
| 150S | ARL | | | 27 | X | | 1 | 12 | X | X | | | X | DIG | 12 + 1 | | | X | Drucker | Tischrechner/%/M+/M– |
| 802MSQ | AL | | 1 | 22 | X | | 1 | 8 | | X | | | | DIG | 8 + 1 | X | X | X | | %/+/– |
| 1011 | ARL | | | 18 | X | | | | X | X | | | X | DIG | 10 + 1 | | | X | | Tischrechner/% |
| 1210N | ARL | | | 23 | X | | 1 | 12 | X | X | | | X | DIG | 12 + 1 | | | X | | Tischrechner/%/M+/M– |
| 1215 | ARL | | | 25 | X | | 1 | 12 | X | X | | | X | DIG | 12 + 1 | | | X | | Tischrechner |
| 8000 | AL | | | 20 | X | | | | | X | | | | DIG | 8 + 1 | | X | | | %/+/– |
| 8005 | AL | | 1 | 22 | X | | | | | X | | | | DIG | 8 + 1 | | X | | | %/+/–/π |
| 8025 | AL | | 1 | 24 | X | | 1 | 8 | | X | | | | DIG | 8 + 1 | | X | | | %/+/–/M+/M– |
| SR-80N | AL | | 14 | 26 | | X | 1 | 8 | | X | | X | | DIG | 8 + 1 | X | X | X | | π/+/–/M+/rad |
| ● **Citizen** | | | | | | | | | | | | | | | | | | | | |
| 121F | ARL | | 1 | 24 | X | | 1 | 12 | X | X | | | X | DIG | 12 + 1 | | | X | | kleiner Tischrechner/%/M+/M– |
| 122LC | ARL | | 1 | 30 | X | | 1 | 10 | X | X | | | X | LCD | 10 + 1 | | X | | | wie 121F |
| 127PII | ARL | | | 25 | X | | 1 | 12 | X | X | | | X | | | | | X | Drucker (12 Stellen) | Tischrechner/%/+/–/M+/M– |
| 128DPII | ARL | | | 25 | X | | 1 | 12 | X | X | | | X | DIG | 12 + 1 | | | X | wie 127PII | wie 127PII |
| 129DP | ARL | | | 30 | X | | 1 | 12 | X | X | | | X | DIG | 12 + 1 | | | X | Drucker | Tischrechner/%/+/–/M+/M– |
| 130P | ARL | | | 24 | X | | 1 | 12 | X | X | | | X | | | | | X | Drucker (12 Stellen) | Tischrechner/%/+/–/M+/M– |

| Hersteller Typ | Rechenlogik | Anzahl der Klammern | Anzahl der Funktionen | Anzahl der Tasten | Tastenbeleg. einfach | Tastenbeleg. mehrfach | Zahl der Speicher | Zahl der Stellen | Komma Fest- | Komma Fließ- | Komma Gleit- | techn.-wiss. | kaufmännisch | Anzeige Art | Anzeige Stellen | Stromversorgung Akku | Stromversorgung Batterie | Stromversorgung Netz | Ergänzungen | Besonderheiten |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| noch | | | | | | | | | | | | | | | | | | | | |
| • **Citizen** | | | | | | | | | | | | | | | | | | | | |
| 131TP | ARL | | | 25 | X | | 1 | 12 | X | X | | | X | | | | | X | Drucker (12 Stellen) | kleiner Tischrechner/% |
| 132CP | ARL | | | 28 | X | | 1 | 10 | X | X | | | X | DIG | 10 + 1 | X | X | X | Drucker | Tischrechner/%/+/_/M+/M– |
| 800D | AL | | 1 | 22 | X | | | | | X | | | | DIG | 8 + 1 | | X | X | | %/π/+/_ |
| 801D | AL | | 1 | 25 | X | | 1 | 8 | | X | | | | DIG | 8 + 1 | | X | X | | %/π/+/_/M+/M– |
| 820SR | AL | | 14 | 26 | | X | 1 | 8 | | X | | X | | DIG | 8 + 1 | X | X | X | | wie CI SR-80N |
| 831RD | AL | | 3 | 28 | X | | 1 | 8 | X | X | | X | | DIG | 8 + 1 | X | X | X | | %/+/_/M+/M– |
| C8 | AL | | 1 | 19 | X | | | | | X | | | | DIG | 8 + 1 | | X | | | % |
| LC8M | AL | | 1 | 24 | X | | 1 | 8 | | X | | | | LCD | 8 + 1 | | X | | | %/+/_/M+/M– |
| LC1000 | AL | | | 23 | X | | | | | X | | | | LCD | 8 + 1 | | X | | | %/+/_/M+/M–/1000 Stunden |
| LC CLOCK | AL | | 1 | 29 | X | | 1 | 8 | | X | | | | LCD | 8 + 1 | | X | | | %/+/_/M+/M–/Stoppuhr/Wecker/5 000 Stunden |
| LC MINI | AL | | 1 | 24 | X | | 1 | 8 | | X | | | | LCD | 8 + 1 | | X | | | %/+/_/M+/M–/6000 Stunden |
| SR I | AL | 2 | 21 | 33 | | X | 1 | 8 | | X | X | X | | DIG | 7 + 2 | X | | X | | π/+/_/M+/M–/Statistik |
| SR II | AL | 2 | 30 | 43 | | X | 5 | 8 | | X | X | X | | DIG | 10 + 2 | X | | X | | wie SR I |
| • **Colex** | | | | | | | | | | | | | | | | | | | | |
| 811A-SL | AL | | | 20 | X | | 1 | 8 | | X | | | | DIG | 8 + 1 | X | | X | | %/M+ |
| CO817MDA | AL | | 13 | 20 | | X | 1 | 8 | | X | | X | | DIG | 8 + 1 | | X | X | | π/+/_/M+/M–/M + $x^2$ |
| CO821ND | AL | | | 20 | X | | | | | X | | | | DIG | 8 + 1 | | X | X | | %/+/_ |
| CO832D | AL | | 1 | 20 | X | | | | | X | | | | DIG | 8 + 1 | | X | X | | % |
| CO835MD | AL | | | 22 | X | | 1 | 8 | | X | | | | DIG | 8 + 1 | | X | X | | %/M+/M– |
| ESR50SL | AL | | 12 | 24 | | X | 1 | 8 | | X | | X | | DIG | 8 + 1 | X | X | X | | π/M+/rad |
| MIDAS | AL/ ALR | | 14 | 20 | | X | 4 | 8 | | X | | | X | DIG | 8 + 1 | X | | X | | %/Δ%/+/_/Finanzen/Statistik |
| • **Commodore** | | | | | | | | | | | | | | | | | | | | |
| 9R23 | AL | | | | | | 1 | 8 | | X | | | | DIG | 8 + 1 | X | | X | | % |
| 9R25 | AL | | 3 | | | | 1 | 8 | | X | | | | DIG | 8 + 1 | X | | X | | % |
| 151P | ARL | | | | | | 1 | | | X | | | | | | | | | Drucker | Tischrechner |
| 152P | ARL | | | | | | 2 | | | X | | | | | | | | | Drucker | Tischrechner |
| 796D | AL | | | 19 | X | | 1 | 8 | | X | | | | LED | 8 + 1 | | X | X | | % |
| 797 | AL | | | 23 | X | | 1 | 8 | | X | | | | LED | 8 + 1 | | X | X | | %/+/_/M+/M– |
| 798 | AL | | 1 | 23 | X | | 1 | 8 | | X | | | | LED | 8 + 1 | | X | X | | %/M+/M– |
| 899D | AL | | | | | | | | | X | | X | | DIG | 8 + 1 | | X | | | |
| 986R | AL | | | 19 | X | | | | | X | | | | DIG | 8 + 1 | X | | X | | %/+/_ |
| 987R | AL | | | 23 | X | | 1 | 8 | | X | | | | DIG | 8 + 1 | X | | X | | %/+/_/M+/M– |
| 1540 | AL | | | | | | | | | | | X | | LED | 10 + 2 | | | X | | Tischrechner |
| 7919D | AL | | 13 | 19 | | X | 1 | 8 | | | X | X | | LED | 7 + 2 | | X | X | | wie CBM |
| F4146R | AL | | 2 | 46 | X | | 2 | 10 | | | X | | X | LED | 12 + 2 | X | | X | | %/+/_/M+/Statistik/Finanztechnik/Δ% |
| F4902 | AL | 1 | 14 | 50 | | X | 1 | 11 | | X | | | X | LED | 11 + 1 | | X | X | | %/Δ%/M+/M–/Statistik/Finanztechnik |
| GL9R23 | AL | | | 23 | X | | 1 | 8 | | X | | | | DIG | 8 + 1 | X | | X | | %/+/_/M+/M– |
| GL9R25 | AL | | 3 | 25 | X | | 1 | 8 | | X | | | | DIG | 8 + 1 | X | | X | | %/M+/M– |
| LC5K1 | AL | | 1 | 24 | X | | 1 | 8 | | X | | | | LCD | 8 + 1 | | X | | | %/M+/M–/5 000 Stunden |
| LC5K3 | AL | | 1 | 24 | X | | 1 | 8 | | X | | | | LCD | 8 + 1 | | X | | | %/Δ%/M+/M–/5 000 Stunden |
| LC5K4 | AL | | | 22 | X | | 1 | 8 | | X | | | | LCD | 8 + 1 | | X | | | %/M+/M– |
| LC43SR | AL | | 18 | 42 | X | | 1 | 8 | | | X | X | | LCD | 10 + 2 | | X | | | %/π/M+/+/_/Statistik/1500 Stunden |
| LC63SR | AL | 2 | 24 | 40 | | X | 1 | 8 | | | X | X | | LCD | 10 + 2 | | X | | | %/π/M+/+/_/rad/Statistik/2500 Stunden |
| LC925 | AL | | 1 | 29 | X | | 1 | 8 | | X | | | | LCD | 8 + 1 | | X | | | %/Δ%/π/M+/M–/3500 Stunden |
| M55 | AL | 2 | ca. 70 | 55 | | X | 6 | 10 | | | X | X | | LED | 12 + 2 | X | | X | | sehr viele mathematische Funktionen |
| N60 | AL | 1 | ca. 60 | 60 | | X | 2 | 10 | | | X | X | | LED | 12 + 2 | X | | X | | diverse Navigationfunktionen |
| S61 | AL | 1 | ca. 40 | 60 | | X | 8 | 10 | | | X | X | | DIG | 12 + 2 | X | | X | | π/+/_/Statistikrechner |
| SR1800 | AL | 1 | 16 | 39 | | X | 2 | 10 | | | X | X | | DIG | 10 + 2 | X | X | X | | %/Δ%/M+/π/+/_/Statistik |
| SR4148R | AL | 1 | 18 | 48 | X | | 2 | 14 | | | X | X | | LED | 12 + 2 | X | | X | | π/+/_/M+/rad/Polar/Statistik |
| SR4190R | AL | 3 | ca. 40 | 49 | | X | 2 | 14 | | | X | X | | LED | 12 + 2 | X | | X | | π/+/_/M+/rad/%/Δ%/Statistik/div. Mathematik-Funktionen/Umrechnungen |

| Hersteller Typ | Rechenlogik | Anzahl der Klammern | Anzahl der Funktionen | Anzahl der Tasten | Tastenbeleg. einfach | Tastenbeleg. mehrfach | Zahl der Speicher | Zahl der Stellen | Komma Fest- | Komma Fließ- | Komma Gleit- | techn.-wiss. | kaufmännisch | Anzeige Art | Anzeige Stellen | Stromversorgung Akku | Stromversorgung Batterie | Stromversorgung Netz | Ergänzungen | Besonderheiten |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| noch | | | | | | | | | | | | | | | | | | | | |
| ● **Commodore** | | | | | | | | | | | | | | | | | | | | |
| SR4912 | AL | 2 | 20 | 49 | X | | 1 | 10 | | | X | X | | LED | 10 + 2 | | X | X | | π/%/+/_ /M+/M x/rad/Statistik |
| SR4921 | UPN | | | | | | | | | | X | X | | LED | 8 + 2 | | | | | |
| SR7919D | AL | | | | | | | | | | X | X | | LED | 7 + 2 | | X | | | |
| SR9190R | AL | 3 | ca. 40 | 49 | | X | 2 | 14 | | | X | X | | LED | 12 + 2 | X | | X | | π/%/Δ %/+/_ /M+/M x/Statistik/Mathematik/ Umrechnungen |
| US5M | | | | | | | | | | | | | | | 12 + 1 | | | | | Tischrechner/%/M+/M– |
| X-24 | AL | | 1 | 24 | X | | 1 | 8 | | X | | | | DIG | 8 + 1 | X | | X | | %/M+/M– |
| ● **COMTEC** | | | | | | | | | | | | | | | | | | | | |
| CMT-333 | AL | | | 23 | X | | 1 | 8 | | X | | | | LCD | 8 + 1 | | X | | | %/M+/M– |
| CMT-555 | AL | | | 22 | X | | | | | X | | | | LCD | 8 + 1 | | X | | | %/Stoppuhr/Wecker |
| CMT-777 | AL | | | 23 | | X | | | | X | | | | LCD | 8 + 1 | | X | | | %/+/_ /M+/Umwandlungen |
| CMT-999 | AL | 2 | 28 | 29 | | X | | | | X | X | X | | LCD | 7 + 2 | | X | | | +/_ /M+/Statistik |
| CMT-411D | AL | | | 29 | X | | 1 | 10 | X | X | | | X | DIG | 10 + 1 | | | X | Drucker | kleiner Tischrechner/%/+/_ /M+/M– |
| CMT-1010 | ARL | | 1 | 26 | X | | 1 | 10 | X | X | | | X | LCD | 10 + 1 | | X | | | kleiner Tischrechner/%/Δ %/+/_ /M+/M– |
| BARON | AL | | 1 | 24 | X | | 1 | 8 | | X | | | | LCD | 8 + 1 | | X | | | %/+/_ /M+/M– |
| DUKE | AL | | 4 | 28 | X | | 1 | 8 | X | X | | | | LCD | 8 + 1 | | X | | | π/%/+/_ /M+/M– |
| ● **Corvus** | | | | | | | | | | | | | | | | | | | | |
| 500 | UPN | | 20 | 30 | | X | 9 | 10 | X | | X | X | | LED | 12 + 2 | X | | X | | π/%/Δ %/+/_ /Hyp./Statistik/rad |
| ● **Electrogamma** | | | | | | | | | | | | | | | | | | | | |
| MP3 | ARL | | | 20 | X | | 1 | 8 | X | X | | | | LED | 8 + 1 | | X | X | | %/M+/M– |
| P3 | ARL | | | 20 | X | | | | X | X | | | | LED | 8 + 1 | | X | X | | %/Division durch 100 |
| W3 | ARL | | 2 | 20 | X | | | | X | X | | | | LED | 8 + 1 | | X | X | | +/_ |
| WP3 | ARL | | 1 | 20 | X | | | | X | X | | | | LED | 8 + 1 | | X | X | | +/_ /% |
| Pythagoras | AL | | 12 | 24 | | X | 1 | 8 | | X | | X | | LED | 8 + 1 | | X | X | | π/e/+/_ /M+/rad |
| ● **Faber-Castell** | | | | | | | | | | | | | | | | | | | | |
| TR1 | AL | | 2 | 21 | | X | 1 | 8 | | X | | X | | LED | 8 + 1 | X | | | | %/M+/Querformat mit abnehmbarem Rechenschieber |
| TR3 | AL | | 12 | 26 | | X | 1 | 8 | | X | | X | | LED | 8 + 1 | X | X | | | wie TR1/rad |
| ● **FACIT** | | | | | | | | | | | | | | | | | | | | |
| 2202 | ARL | | | 23 | X | | | | X | X | | | X | | | | | X | Drucker (12 Stellen) | kleiner Tischrechner/% |
| ● **FORDS** | | | | | | | | | | | | | | | | | | | | |
| SC85M | AL | | 12 | 26 | | X | 1 | 8 | | X | | X | | DIG | 8 + 1 | | X | X | | π/ +/_ /M+/rad |
| VIP121M | AL | | | 23 | X | | 1 | 12 | X | X | | | X | DIG | 12 + 1 | | | X | | Tischrechner/% |
| ● **General Corp.** | | | | | | | | | | | | | | | | | | | | |
| T2002P | ARL | | | 25 | X | | | | X | X | | | X | | | | | X | Drucker (12 Stellen) | Tischrechner |
| T2003P | ARL | | | 24 | X | | | | X | X | | | X | | | | | X | wie 2002P | Tischrechner/% |
| T2003PD | ARL | | | 24 | X | | | | X | X | | | X | DIG | 12 + 1 | | | X | wie 2003P | wie 2003P |
| T2104 | ARL | | 1 | 24 | X | | 1 | 12 | X | X | | | X | DIG | 12 + 1 | | | X | | Tischrechner/M+/M– |
| T2111P | ARL | | 1 | 32 | X | | 1 | 12 | X | X | | | X | | | | | X | Drucker (12 Stellen) | Tischrechner/%/+/_ /M+/M– |
| T2112P | ARL | | 1 | 32 | X | | 1 | 12 | X | X | | | X | | | | | X | wie 2111P | wie 2111P |
| T8105 | AL | | 3 | 25 | X | | 1 | 8 | | X | | | | LCD | 8 + 1 | | X | | | %/+/_ /M+ |
| ● **Gutag** | | | | | | | | | | | | | | | | | | | | |
| MINI004 | AL | | 1 | 24 | X | | 1 | 8 | | X | | | | LED | 8 + 1 | X | | | | %/M+/M– |
| ● **HANIMEX** | | | | | | | | | | | | | | | | | | | | |
| 200 | AL | | | 18 | X | | | | | X | | | | DIG | 8 + 1 | | X | X | | % |

| Hersteller Typ | Rechenlogik | Anzahl der Klammern | Anzahl der Funktionen | Anzahl der Tasten | Tastenbeleg. einfach | Tastenbeleg. mehrfach | Zahl der Speicher | Zahl der Stellen | Komma Fest- | Komma Fließ- | Komma Gleit- | techn.-wiss. | kaufmännisch | Anzeige Art | Anzeige Stellen | Stromversorgung Akku | Stromversorgung Batterie | Stromversorgung Netz | Ergänzungen | Besonderheiten |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| noch | | | | | | | | | | | | | | | | | | | | |
| • **HANIMEX** | | | | | | | | | | | | | | | | | | | | |
| 200M | AL | | | 20 | X | | 1 | 8 | | X | | | | DIG | 8 + 1 | | X | X | | %/M+ |
| 1500DP | ARL | | 1 | 34 | X | | 2 | 12 | X | X | | | X | DIG | 12 + 1 | | | X | Drucker | Tischrechner/%/+/_ /M+/M– |
| BC900 | AL | | | 19 | X | | | | | X | | | | DIG | 8 + 1 | | X | X | | % |
| BC999 | AL | | 2 | 22 | X | | | | | X | | | | DIG | 8 + 1 | | X | X | | % |
| BCM368 | AL | | | 23 | X | | 1 | 8 | | X | | | | DIG | 8 + 1 | | X | X | | %/M+/M– |
| BCM368R | AL | | 1 | 23 | X | | 1 | 8 | | X | | | | DIG | 8 + 1 | | X | X | | %/+/_ /M+/M– |
| BCM378 | AL | | 3 | 26 | X | | 1 | 8 | | X | | | | DIG | 8 + 1 | | X | X | | %/M+/M– |
| BCM901 | AL | | | 23 | X | | 1 | 8 | | X | | | | DIG | 8 + 1 | | X | X | | %/+/_ /M+/M– |
| ESR80 | AL | | 4 | 23 | X | | 1 | 8 | | X | | | | DIG | 8 + 1 | | X | X | | % |
| ESR90SN | AL | | 4 | 23 | X | | 1 | 8 | | | X | | | DIG | 7 + 2 | | X | X | | % |
| ESR100 | AL | | 12 | 26 | | X | 1 | 8 | | X | | X | | DIG | 8 + 1 | | X | X | | π/+/_ /M+/rad |
| ESR1010SN | AL | | 12 | 26 | | X | 1 | 8 | | X | X | X | | DIG | 7 + 2 | | X | X | | wie ESR100 |
| ESR388 | AL | 1 | 3 | 29 | X | | 1 | 8 | | X | | | | DIG | 8 + 1 | | X | X | | π/+/_ /%/M+/M– |
| ESR398 | AL | 2 | 14 | 27 | | X | 1 | 8 | | X | X | X | | DIG | 7 + 2 | | X | X | | π/+/_ /%/M+/M–/rad |
| LC8-1 | AL | | | 21 | X | | 1 | 8 | | X | | | | LCD | 8 + 1 | | X | | | %/M+/M– |
| LC640 | AL | | 1 | 18 | X | | 1 | 8 | | X | | | | LCD | 8 + 1 | | X | | | %/M+/M–/800 Stunden |
| LC774 | AL | | 4 | 27 | X | | 1 | 8 | | X | | | | LCD | 8 + 1 | | X | | | π/%/M+/M–/1500 Stunden |
| LC776 | AL | | | 29 | X | | 1 | 8 | | X | | | | LCD | 8 + 1 | | X | | | %/M+/M–/Stoppuhr/Kalender/Wecker/ 5000 Stunden |
| LC777S | AL | | 1 | 25 | X | | 1 | 8 | | X | | | | LCD | 8 + 1 | | X | | | %/+/_ /M+/M–/1200 Stunden |
| LC780 | AL | 2 | 30 | 40 | | X | 1 | 8 | | X | X | X | | LCD | 10 + 2 | | X | | | π/+/_ /Δ %/M+/Statistik/500 Stunden |
| LCD8M | AL | | | 23 | X | | 1 | 8 | | X | | | | LCD | 8 + 1 | | X | | | %/+/_ /M+/M– |
| micro | AL | | | 22 | X | | 1 | 8 | | X | | | | LCD | 8 + 1 | | X | | | %/M+/M–/1500 Stunden |
| VP8SR | AL | | 10 | 19 | | X | 1 | 8 | | X | X | | | LED | 7 + 2 | | X | X | | %/M+/M– |
| • **HEATHKIT** | | | | | | | | | | | | | | | | | | | | |
| IC-2008 | AL | | | 20 | X | | | | X | X | | | | LED | 8 + 1 | | | X | | Tischrechner/+/_ |
| **Helec** | | | | | | | | | | | | | | | | | | | | |
| FR6032 | AL | 2 | 14 | 28 | | X | 1 | 10 | | X | | | | DIG | 12 + 2 | X | X | X | | +/_ /M+/M–/M x/M ÷/rad/Querformat |
| GT530 | ARL | | | 19 | X | | | | X | X | | | | LED | 8 + 1 | | X | X | | % |
| GT531 | AL | | 12 | 26 | | X | 1 | 8 | | X | | X | | LED | 8 + 1 | | X | X | | π/+/_ /rad |
| GT532 | AL | 2 | 14 | 25 | | X | 1 | 8 | | X | X | X | | DIG | 10 + 2 | | X | X | | +/_ /M+/M–/M x/M ÷/rad/π |
| GT534 | ARL | | | 19 | X | | | | X | X | | | | LED | 8 + 1 | | X | X | | %/+/_ |
| GT535 | ARL | | | 19 | X | | | | X | X | | | | LED | 10 + 1 | | X | X | | +/_ |
| GT538 | ARL | | 1 | 19 | X | | | | X | X | | | | LED | 8 + 1 | | X | X | | % |
| GT539 | ARL | | | 20 | | | 1 | 8 | | X | | | X | LED | 8 + 1 | | X | X | | %/M+ |
| • **Hermes Precisa** | | | | | | | | | | | | | | | | | | | | |
| 1010 | AL | | | 28 | | | 1 | 10 | X | X | | | X | LED | 10 | X | X | X | Drucker | Tischrechner/%/M+/M– |
| • **Hewlett Packard** | | | | | | | | | | | | | | | | | | | | |
| HP-10 | ARL | | | 24 | | X | 1 | 10 | | X | | | X | LED | 10 | X | | | Drucker | %/M+/M– |
| HP-21 | UPN | | 13 | 30 | | X | 1 | 10 | X | | X | X | | LED | 10 + 2 | X | | | | π/M+/M–/rad/M x/M ÷/+/_ |
| HP-22 | UPN | | 8 | 30 | | X | 15 | 10 | | | X | | X | LED | 10 + 2 | X | | | | %/Δ % |
| HP-27 | UPN | | 21 | 30 | | X | 15 | 10 | X | | X | X | X | LED | 10 + 2 | X | | | | π/%/Δ %/rad/Statistik |
| HP-31E | UPN | | 13 | 30 | | X | 4 | 10 | X | | X | X | | LED | 9 + 2 | X | | | | %/rad/Polar/π/Umwandlungen |
| HP-32E | UPN | | 23 | 30 | | X | 15 | 10 | X | | X | X | | LED | 9 + 2 | X | | | | %/rad/Polar/π/Δ %/Umwandlungen/Statistik |
| HP-35 | UPN | | 10 | 35 | X | | 1 | 10 | | | X | X | | LED | 12 + 2 | X | | | | π/+/_ |
| HP-37E | UPN | | 9 | 30 | | X | 7 | 10 | X | | X | | X | LED | 10 + 1 | X | | | | %/Δ %/Zins/Statistik |
| HP-45 | UPN | | 16 | 35 | | X | 9 | 10 | X | | X | X | | LED | 12 + 2 | X | | | | π/%/Δ %/rad/Polar/Umwandlungen |
| HP-46 | UPN | | 16 | 42 | | X | 9 | 10 | X | | X | X | | LED | 12 + 2 | X | | | Drucker | Tischrechner/π/%/Δ %/rad/Polar/Umwandlungen |
| HP-70 | UPN | | 1 | 35 | X | | 5 | 10 | X | | X | | X | LED | 12 + 2 | X | | | | %/Δ %/M+/+/_ |
| HP-80 | UPN | | 5 | 35 | | X | 1 | 10 | X | | X | | X | LED | 12 + 2 | X | | | | %/Δ %/Kalender |
| HP-81 | UPN | | 5 | 43 | | X | 20 | 10 | X | | X | | X | LED | 12 + 2 | X | | | Drucker | Tischrechner/wie HP-80 |
| HP-91 | UPN | | 19 | 42 | | X | 16 | 10 | X | | X | X | | LED | 12 + 2 | X | | | Drucker | Tischrechner/%/Δ %/π/rad/Polar/Statistik |
| HP-92 | UPN | | 10 | 42 | | X | 38 | 10 | | | | | X | LED | 12 + 2 | X | | | | Tischrechner/Kalender/%/Δ %/Statistik |

| Hersteller Typ | Rechenlogik | Anzahl der Klammern | Anzahl der Funktionen | Anzahl der Tasten | Tastenbeleg. einfach | Tastenbeleg. mehrfach | Zahl der Speicher | Zahl der Stellen | Komma Fest- | Komma Fließ- | Komma Gleit- | techn.-wiss. | kaufmännisch | Anzeige Art | Anzeige Stellen | Stromversorgung Akku | Stromversorgung Batterie | Stromversorgung Netz | Ergänzungen | Besonderheiten |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Hitachi** | | | | | | | | | | | | | | | | | | | | |
| KK221B | ARL | | | 20 | X | | 1 | 12 | | X | | | | DIG | 8 + 1 | X | X | X | | %/kleiner Tischrechner |
| **Ibico** | | | | | | | | | | | | | | | | | | | | |
| 052 | AL | | | 22 | X | | 1 | 8 | | X | | | | LCD | 8 + 1 | | X | | | %/M+/M– |
| 053 | AL | | 1 | 22 | X | | 1 | 8 | | X | | | | LCD | 8 + 1 | | X | | | %/M+/M– |
| 055 | AL | | | 22 | | X | 1 | 8 | | X | | | | LCD | 8 + 1 | | X | | | wie 053/Stoppuhr/Wecker/Kalender |
| 062 | AL | | 1 | 24 | X | | 1 | 8 | | X | | | | LCD | 8 + 1 | | X | | | %/+/–/M+/M–/1 000 Stunden |
| 066 | AL | | 1 | 24 | X | | 1 | 8 | | X | | | | LCD | 8 + 1 | | X | | | wie 062 |
| 075 | AL | | | 24 | | X | | | | | | | | LCD | 8 + 1 | | X | | | %/+/–/M+/M–/Stoppuhr/Kalender/Wecker |
| 083 | AL | | | 19 | X | | | | | X | | | | DIG | 8 + 1 | | X | X | | % |
| 087 | AL | | | 20 | X | | 1 | 8 | X | X | | | | DIG | 8 + 1 | | X | X | | % |
| 088 | AL | | 3 | 24 | X | | 1 | 8 | | X | | | | DIG | 8 + 1 | | X | X | | %/+/– |
| 093 | AL | | 12 | 27 | | X | 1 | 8 | | X | | X | | DIG | 8 + 1 | | X | X | | +/–/π/M+/rad |
| 096 | AL | 2 | 26 | 40 | | X | 1 | 8 | | | X | X | | LCD | 10 + 2 | | X | | | π/+/–/M+/rad/Hyp./Statistik |
| 1202 | ARL | | | 26 | X | | | | X | X | | | X | | | | | X | Drucker (12 Stellen) | Tischrechner/%/+/– |
| 1217 | ARL | | | 29 | X | | 1 | 12 | X | X | | | X | DIG | 12 + 1 | | | X | Drucker | Tischrechner/% |
| 2500 | ARL | | | 28 | X | | 1 | 12 | X | X | | | X | DIG | 12 + 1 | | | X | Drucker | Tischrechner/%/M+/M– |
| **INTERTON** | | | | | | | | | | | | | | | | | | | | |
| PC4009 | AL | | | 18 | X | | | | | X | | | | LED | 8 + 1 | | X | X | | % |
| PC4019 | AL | | 3 | 27 | X | | 1 | 8 | | X | | | | LED | 8 + 1 | | X | X | | %/+/–/M+/M– |
| PC4044 | AL | 2 | 14 | 20 | | X | 1 | 10 | | | X | X | | LED | 12 + 2 | | X | X | | π/+/–/M+/M–/rad |
| PC5009 | AL | | | 19 | X | | | | | X | | | | LED | 8 + 1 | | X | X | | %/+/– |
| PC5019 | AL | | 1 | 24 | X | | 1 | 8 | | X | | | | LED | 8 + 1 | | X | X | | %/M+/M– |
| PC5030 | AL | 2 | 14 | 24 | X | | 1 | 8 | | | X | X | | LED | 7 + 2 | | X | X | | π/+/–/M+/M–/rad |
| PC6010 | AL | | 3 | 23 | X | | | | X | X | | | | DIG | 8 + 1 | X | | X | | %/+/– |
| PC6020 | AL | | 3 | 30 | X | | 1 | 8 | X | X | | | | DIG | 8 + 1 | X | | X | | %/+/–/M+/M– |
| PC6040 | AL | 2 | 15 | 40 | X | | 1 | 8 | | | X | X | | DIG | 10 + 2 | X | | X | | π/+/–/M+/rad |
| PC6050 | AL | 2 | 15 | 35 | | X | 1 | 8 | | | X | | X | DIG | 10 + 2 | X | | X | | %/+/–/Δ%/Finanzrechner |
| VIP10 | AL | | | 20 | X | | | | X | X | | | | LED | 10 + 1 | | X | X | | %/+/– |
| **ITOH** | | | | | | | | | | | | | | | | | | | | |
| 1217 | ARL | | 1 | 25 | X | | 1 | 12 | X | X | | | X | DIG | 12 + 1 | | | X | | Tischrechner/% |
| 1426 | ARL | | 1 | 25 | X | | 2 | 14 | X | X | | | X | DIG | 14 + 1 | | | X | | wie 1217 |
| CI25S | ARL | | | 30 | X | | | | X | X | | | X | | | | | X | Drucker (12 Stellen) | Tischrechner/% |
| CI35S | ARL | | | 30 | X | | | | X | X | | | X | DIG | 12 + 1 | | | X | wie 25S | wie 25S |
| CI45S | ARL | | | 30 | X | | 1 | 12 | X | X | | | X | | | | | X | wie 25S | wie 25S/M+/M– |
| CI95S | ARL | | | 30 | X | | 1 | 12 | X | X | | | X | DIG | 12 + 1 | | | X | wie 45S | wie 45S |
| CI101P | AL | | | 29 | X | | 1 | 10 | X | X | | | X | | | | | X | Drucker (10 Stellen) | kleiner Tischrechner/M+/M– |
| CI101PD | AL | | | 29 | X | | 1 | 10 | X | X | | | X | DIG | 10 + 1 | | | X | wie 101P | wie 101P |
| LC2500 | AL | | 1 | 24 | X | | 1 | 8 | | X | | | | LCD | 8 + 1 | | X | | | %/+/–/M+/M– |
| **Jet-Electronics** | | | | | | | | | | | | | | | | | | | | |
| Sabtronics SC8W | AL | | 1 | 31 | X | | | | | X | | | | LCD | 8 + 1 | | X | | | %/M+/M–/Wecker/Stoppuhr/Kalender/ 13 000 Stunden |
| **Karstadt** | | | | | | | | | | | | | | | | | | | | |
| EP20 | AL | | | 19 | X | | | | | X | | | | DIG | 8 + 1 | | X | X | | % |
| MR40 | AL | | 1 | 22 | X | | 1 | 8 | | X | | | | DIG | 8 + 1 | | X | X | | %/+/– |
| PD60 | ARL | | | 25 | X | | | | X | X | | | | DIG | 12 + 1 | | | X | Drucker | Tischrechner/%/M+/M– |
| **Keystone** | | | | | | | | | | | | | | | | | | | | |
| 2010 | AL | | | 20 | X | | | | | X | | | | DIG | 8 + 1 | | X | | | %/+/– |
| 2030 | AL | | | 24 | X | | 1 | 8 | | X | | | | DIG | 8 + 1 | X | | X | | %/+/–/M+/M– |
| 2040 | AL | | 1 | 22 | X | | 1 | 8 | | X | | | | DIG | 8 + 1 | X | | X | | %/+/– |
| 2050 | AL | | 12 | 30 | X | | 1 | 8 | | X | | X | | DIG | 8 + 1 | X | | X | | π/+/–/M+/M– |

| Hersteller Typ | Rechenlogik | Anzahl der Klammern | Anzahl der Funktionen | Anzahl der Tasten | Tastenbeleg. einfach | Tastenbeleg. mehrfach | Zahl der Speicher | Zahl der Stellen | Komma Fest- | Komma Fließ- | Komma Gleit- | techn.-wiss. | kaufmännisch | Anzeige Art | Anzeige Stellen | Stromversorgung Akku | Stromversorgung Batterie | Stromversorgung Netz | Ergänzungen | Besonderheiten |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **KOVAC** | | | | | | | | | | | | | | | | | | | | |
| 809ESR | AL | | 12 | 26 | | X | 1 | 8 | | X | | X | | DIG | 8 + 1 | | X | | | π/rad |
| 810F | AL | | 13 | 20 | | X | | | X | X | | | X | DIG | 8 + 1 | | X | X | | %/Δ%/+/− |
| 811ESR | AL | 2 | 14 | 25 | | X | 1 | 8 | | | X | X | | DIG | 10 + 2 | X | | X | | π/M+/M−/rad |
| 868MR | AL | | 3 | 24 | X | | 1 | 8 | X | X | | | | DIG | 8 + 1 | | X | X | | %/+/− |
| 878M | AL | | | 21 | X | | 1 | 8 | X | X | | | | DIG | 8 + 1 | | X | | | %/+/− |
| F-8PR | AL | | 1 | 20 | X | | | | | X | | | | DIG | 8 + 1 | | X | X | | % |
| KP-1212M | ARL | | | 25 | X | | 1 | 12 | X | X | | | | | | | | X | Drucker (12 Stellen) | Tischrechner/%/M+/M− |
| LC-116 | AL | | | 23 | | X | 1 | 8 | | X | | | | LCD | 8 + 1 | | X | | | M+/M−/Uhr/Wecker |
| PD-15 | AL | | | 28 | X | | 1 | 12 | X | X | | | | DIG | 12 + 1 | X | X | X | Drucker | kleiner Tischrechner/%/M+/M− |
| PM-1200M | ARL | | | 27 | X | | 1 | 12 | X | X | | | | DIG | 12 + 1 | | | X | | Tischrechner/%/M+/M− |
| P80 | AL | | 1 | 20 | X | | | | | X | | | | DIG | 8 + 1 | | X | X | | % |
| P81 | AL | | 1 | 23 | X | | 1 | 8 | | X | | | | DIG | 8 + 1 | | X | X | | %/+/−/M+/M− |
| P82 | AL | 2 | 13 | 25 | | X | 1 | 8 | | X | | X | | DIG | 7 + 2 | | X | X | | π/+/− |
| P83 | AL | | 3 | 25 | X | | | | | X | | | | DIG | 8 + 1 | | X | X | | π/+/−/%/M+ |
| P85 | AL | | 18 | 35 | X | | 1 | 10 | | | X | X | | DIG | 12 + 2 | | X | X | | π/+/−/M+/rad |
| SM1210M | ARL | | | 24 | X | | 1 | 12 | X | X | | | X | DIG | 12 + 1 | X | | X | | % |
| **Litronix** | | | | | | | | | | | | | | | | | | | | |
| 1102 | ARL | | | 18 | X | | | | | X | | | | LED | 8 + 1 | | X | X | | % |
| 1601 | AL | | 1 | 22 | X | | | | | X | | | | LED | 8 + 1 | | X | X | | %/+/− |
| 1602 | ARL | | | 24 | X | | 1 | 8 | | X | | | | LED | 8 + 1 | | X | X | | %/M+/M− |
| 1603 | AL | | 2 | 24 | X | | 1 | 8 | | X | | | | LED | 8 + 1 | | X | X | | %/M+/M−/M x/M ÷ |
| 2200 | AL | | | 25 | | X | 1 | 8 | | X | | | | LED | 8 + 1 | | X | X | | %/+/−/M+/M−/Umwandlungen |
| 2210 | AL | | 3 | 25 | X | | | | | X | | | | LED | 8 + 1 | | X | X | | π/%/+/− |
| 2230 | AL | | 1 | 25 | X | | 1 | 8 | | X | | | | LED | 8 + 1 | | X | X | | %/+/−/M+/M− |
| 2260 | AL | 2 | 2 | 25 | X | | 1 | 8 | | X | X | | | LED | 7 + 2 | | X | X | | π/+/− |
| 2280 | AL | | | 25 | X | | 2 | 8 | | X | | | | LED | 8 + 1 | | X | X | | %/M+/M− |
| **Litton** | | | | | | | | | | | | | | | | | | | | |
| RC84 | AL | | 1 | 20 | X | | | | | X | | | | DIG | 8 + 1 | X | X | X | | % |
| RC87 | AL | | 1 | 22 | X | | 1 | 8 | | X | | | | DIG | 8 + 1 | X | X | X | | %/+/− |
| RSC40 | AL | | 3 | 24 | | X | 1 | 8 | | X | | | | DIG | 8 + 1 | X | X | X | | π/%/+/−/M+/M− |
| RSC60 | AL | | 11 | 26 | | X | 1 | 8 | | X | | X | | DIG | 8 + 1 | X | X | X | | π/+/−/M+ |
| RSC65 | AL | 2 | 14 | 25 | | X | 1 | 8 | | | X | X | | DIG | 10 + 2 | X | X | X | | π/+/−/M+/M−/M x/M ÷ |
| **Logitech** | | | | | | | | | | | | | | | | | | | | |
| LC-10P | AL | | | 27 | X | | 1 | 10 | X | X | | | | DIG | 10 + 1 | X | X | X | Drucker | kleiner Tischrechner/%/+/−/M+/M− |
| LC-18K | AL | | 1 | 24 | X | | 1 | 8 | | X | | | | LCD | 8 + 1 | | X | | | %/M+/M−/2000 Stunden |
| LC-30U | AL | | 1 | 23 | X | | 1 | 8 | | X | | | | LCD | 8 + 1 | | X | | | wie 18K |
| LC-80S | AL | 2 | 40 | 39 | | X | 1 | 8 | | | X | X | | LCD | 10 + 2 | | X | | | π/+/−/M+/M−/Statistik |
| LC-220 | ARL | | | 32 | X | | 2 | 12 | X | X | | | X | DIG | 12 + 1 | X | X | X | | kleiner Tischrechner |
| **MARENO** | | | | | | | | | | | | | | | | | | | | |
| Speech+ | AL | | 1 | 24 | X | | 1 | 8 | | X | | | | LED | 8 + 1 | | X | | | %/Sprachausgabe |
| **Matsushita Panasonic** | | | | | | | | | | | | | | | | | | | | |
| JE-8210U | AL | | 1 | 25 | | X | 1 | | | X | | | | DIG | 8 + 1 | | X | | | π/%/M+/M−/Umrechnungen/Längenmessungen |
| **MBO** | | | | | | | | | | | | | | | | | | | | |
| 12PD | AL | | | 28 | X | | 2 | 11 | X | X | | | X | DIG | 11 + 1 | | X | X | Drucker | Tischrechner/%/M+/M−/+/− |
| 830ND | AL | | | 20 | X | | | | | X | | | | DIG | 8 + 1 | | X | X | | %/+/− |
| 833ND | AL | | 1 | 20 | X | | | | | X | | | | DIG | 8 + 1 | | X | X | | % |
| 837MD | AL | | | 22 | X | | 1 | 8 | | X | | | | DIG | 8 + 1 | | X | X | | %/M+/M− |
| 838MD | AL | | 3 | 23 | X | | 1 | 8 | X | X | | | | DIG | 8 + 1 | | X | X | | % |
| 2002W | AL | | 1 | 24 | X | | 1 | 8 | | X | | | | LCD | 8 + 1 | | X | | | %/+/−/M+/M−/Quarzuhr/2000 Stunden |
| 2003W | AL | | 1 | 24 | X | | 1 | 8 | | X | | | | LCD | 8 + 1 | | X | | | wie 2002W plus Wecker |
| 3000 | AL | | | 24 | X | | 1 | 8 | | X | | | | DIG | 8 + 1 | | X | | | %/+/−/M+/M− |

| Hersteller Typ | Rechenlogik | Anzahl der Klammern | Anzahl der Funktionen | Anzahl der Tasten | Tastenbeleg. einfach | Tastenbeleg. mehrfach | Zahl der Speicher | Zahl der Stellen | Komma Fest- | Komma Fließ- | Komma Gleit- | techn.-wiss. | kaufmännisch | Anzeige Art | Anzeige Stellen | Stromversorgung Akku | Stromversorgung Batterie | Stromversorgung Netz | Ergänzungen | Besonderheiten |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| noch | | | | | | | | | | | | | | | | | | | | |
| ● **MBO** | | | | | | | | | | | | | | | | | | | | |
| Alpha II | AL | 3 | 24 | 40 | | X | 3 | 10 | | | X | X | | LED | 12 + 2 | X | | X | | $\pi$/$^+$/_ /M+/rad/Statistik |
| Alpha 10 | AL | | 13 | 20 | | X | 1 | 8 | | X | X | X | | DIG | 7 + 2 | | X | X | | $\pi$/$^+$/_ /M+/M–/M + $x^2$ |
| Alpha 50 | AL | 1 | 3 | 30 | X | | 1 | 8 | | X | | | | DIG | 8 + 1 | | X | X | | %/$^+$/_ /M+/M– |
| Alpha 51 | AL | 2 | 14 | 36 | X | | 1 | 8 | | X | X | X | | DIG | 7 + 2 | | X | X | | $\pi$/$^+$/_ /M+/rad |
| Alpha 52 | AL | 2 | 14 | 37 | | X | 1 | 8 | | | X | X | | DIG | 10 + 2 | | X | X | | $\pi$/$^+$/_ /M+/rad |
| Alpha 70 | AL | | 15 | 20 | | X | 1 | 8 | | X | | | X | DIG | 8 + 1 | | X | X | | $^+$/_ /%/$\Delta$%/Finanzrechner |
| Alpha 1000-1 | AL | 2 | 14 | 40 | X | | 1 | 8 | | X | X | X | | DIG | 7 + 2 | X | X | X | | $\pi$/$^+$/_ /%/M+/M–/rad |
| Alpha 2000 | AL | 2 | 23 | 48 | X | | 1 | 10 | | | X | X | | DIG | 10 + 2 | X | X | X | | $\pi$/$^+$/_ /M+/Hyp./rad |
| Alpha 3000 | AL | 2 | 23 | 43 | | X | 1 | 10 | | | X | X | | DIG | 10 + 2 | X | X | X | | $\pi$/$^+$/_ /M+/M–/M x/M ÷/Hyp./rad/Statistik |
| Akkord II | AL | | 3 | 28 | X | | 1 | 8 | | X | | | | DIG | 8 + 1 | | X | | | $\pi$/$^+$/_ /%/M+/M– |
| Classic I | AL | | 1 | 22 | X | | | | | X | | | | DIG | 8 + 1 | X | X | X | | $\pi$/$^+$/_ /% |
| Classic II | AL | | 1 | 24 | X | | 1 | 8 | | X | | | | DIG | 8 + 1 | X | X | X | | $\pi$/$^+$/_ /%/M+/M– |
| Classic III | AL | | 3 | 28 | X | | 1 | 8 | | X | | | | DIG | 8 + 1 | X | X | X | | $\pi$/$^+$/_ /%/M+/M– |
| Concorde I | AL | | | 19 | X | | | | | X | | | | DIG | 8 + 1 | X | X | X | | % |
| Concorde II | AL | | 1 | 22 | X | | | | | X | | | | DIG | 8 + 1 | X | X | X | | $\pi$/$^+$/_ /% |
| Concorde III | AL | | 1 | 22 | X | | 1 | 8 | | X | | | | DIG | 8 + 1 | X | X | X | | %/$^+$/_ |
| Concorde IV | AL | | 3 | 23 | | X | | | | X | | | | DIG | 8 + 1 | X | X | X | | $\pi$/$^+$/_ /%/M+/M– |
| de Luxe I | AL | | | 20 | X | | | | | X | | | | DIG | 8 + 1 | | X | X | | %/$^+$/_ |
| de Luxe II | AL | | 1 | 22 | X | | | | | X | | | | DIG | 8 + 1 | | X | X | | $\pi$/$^+$/_ /% |
| de Luxe III | AL | | 1 | 22 | X | | 1 | 8 | | X | | | | DIG | 8 + 1 | | X | X | | %/$^+$/_ |
| Exakt 10 | AL | | 3 | 23 | X | | | | | X | | | | DIG | 8 + 1 | | X | X | | $\pi$/$^+$/_ /% |
| Exakt 20 | AL | | 1 | 23 | X | | 1 | 8 | | X | | | | DIG | 8 + 1 | | X | X | | %/M+/M– |
| Exakt 30 | AL | | 4 | 28 | X | | 1 | 8 | | X | | | | DIG | 8 + 1 | | X | X | | $\pi$/$^+$/_ /%/M+/M– |
| Exakt 40 | AL | | 20 | 38 | X | | 1 | 10 | | | X | X | | DIG | 10 + 2 | | X | X | | $\pi$/$^+$/_ /M+/rad/Hyp. |
| Formel I | AL | | | 22 | X | | 1 | 8 | | X | | | | LCD | 8 + 1 | | X | | | %/M+/M–/3 000 Stunden |
| Formel II | AL | | 1 | 25 | X | | 1 | 8 | | X | | | | LCD | 8 + 1 | | X | | | %/$^+$/_ /M+/M–/ 3 000 Stunden |
| Formel III | AL | | 4 | 27 | X | | | | | X | | | | LCD | 8 + 1 | | X | | | $\pi$/%/M+/M–/3 000 Stunden |
| Formel IV | AL | | | 29 | X | | 1 | 8 | | X | | | | LCD | 8 + 1 | | X | | | %/M+/M–/Stoppuhr/3 000 Stunden |
| Formel 20 | AL | 2 | 27 | 35 | | X | 1 | 8 | | | X | X | | LCD | 7 + 2 | | X | | | $\pi$/$^+$/_ /M+/rad/Statistik/2 000 Stunden |
| Formel 21 | AL | 2 | 30 | 40 | | X | 1 | 10 | | | X | X | | LCD | 10 + 2 | | X | | | $\pi$/$^+$/_ /M+/rad/Polar/Statistik/2 000 Stunden |
| FX102 | AL | | 20 | 38 | | X | 1 | 10 | | | X | X | | DIG | 10 + 2 | | X | | | $\pi$/$^+$/_ /M+/M–/rad/Statistik |
| LC811 | AL | | | 23 | X | | 1 | 8 | | X | | | | LCD | 8 + 1 | | X | | | %/$^+$/_ /M+/M–/1 200 Stunden |
| LC820 | AL | | 1 | 24 | X | | 1 | 8 | | X | | | | LCD | 8 + 1 | | X | | | %/$^+$/_ /M+/M–/2 000 Stunden |
| LC2000 | AL | | 1 | 24 | X | | 1 | 8 | | X | | | | LCD | 8 + 1 | | X | | | %/$^+$/_ /M+/M–/2 000 Stunden |
| LC Alpha 4000 | AL | 2 | 18 | 35 | | X | 1 | 8 | | X | X | X | | LCD | 7 + 2 | | X | | | $\pi$/$^+$/_ /M+/rad/Statistik/2 000 Stunden |
| LC Alpha 5000 | AL | 2 | 22 | 35 | | X | 1 | 8 | | X | X | X | | LCD | 10 + 2 | | X | | | wie 4000, zusätzliche Hyp./Polar/mehr Statistik |
| Monarch 20 | AL | | 1 | 24 | X | | 1 | 8 | | X | | | | DIG | 8 + 1 | X | X | X | | $\pi$/$^+$/_ /%/M+/M– |
| Monarch 30 | AL | | 3 | 28 | X | | 1 | 8 | | X | | | | DIG | 8 + 1 | X | X | X | | $\pi$/$^+$/_ /%/M+/M– |
| Monarch CX40 | AL | 2 | 14 | 40 | X | | 1 | 8 | | X | | X | | DIG | 8 + 1 | X | X | X | | $\pi$/$^+$/_ /%/M+/M–/rad |
| Monarch CX50 | AL | 2 | 24 | 48 | X | | 1 | 8 | | | X | X | | DIG | 10 + 2 | X | X | X | | $\pi$/$^+$/_ /M+/rad/Hyp. |
| SL I | AL | | 1 | 20 | X | | | | | X | | | | DIG | 8 + 1 | | X | X | | % |
| SL II | AL | | 1 | 24 | X | | 1 | 8 | | X | | | | DIG | 8 + 1 | | X | X | | $\pi$/$^+$/_ /%/M+/M– |
| SL III | AL | | 3 | 24 | | X | 1 | 8 | | X | | | | DIG | 8 + 1 | | X | X | | $\pi$/$^+$/_ /%/M+/M– |
| TR12M | ARL | | 1 | 25 | X | | 1 | 12 | X | X | | | X | DIG | 12 + 1 | | X | X | | Tischrechner/%/$^+$/_ /M+/M– |
| TR81M | AL | | 1 | 24 | X | | 1 | 8 | | X | | | | DIG | 8 + 1 | | X | X | | Tischrechner/$\pi$/$^+$/_ /%/M+/M– |
| TRS1000 | ARL | | | 28 | X | | 1 | 12 | X | X | | | | | | | | X | Drucker (12 Stellen) | Tischrechner/$^+$/_ /M+/M– |
| TRS1300PCR | ARL | | 1 | 30 | X | | 2 | 12 | X | X | | | X | DIG | 12 + 1 | | | X | Drucker | Tischrechner mit Registrierkasse |
| TRS1450PD | ARL | | | 32 | X | | 2 | 12 | X | X | | | X | DIG | 12 + 1 | | | X | Drucker | Tischrechner/%/$^+$/_ /M+/M– |
| ● **Melcor** | | | | | | | | | | | | | | | | | | | | |
| SC-615 | AL | 2 | 12 | 40 | X | | 1 | 8 | | X | X | X | | LED | 10 + 2 | X | | X | | $\pi$/$^+$/_ /M+ |
| SC-650 | AL | 2 | 22 | 40 | | X | 3 | 10 | | X | X | X | | LED | 12 + 2 | X | | X | | $\pi$/$^+$/_ /M+/rad/Statistik |
| ● **NASCO** | | | | | | | | | | | | | | | | | | | | |
| 12PD | ARL | | | 29 | X | | 1 | 12 | X | X | | | X | DIG | 12 + 1 | | | X | Drucker | Tischrechner/%/$^+$/_ /M+/M– |
| 101PDS | ARL | | | 26 | X | | | | X | X | | | X | DIG | 10 + 1 | | | X | Drucker | Tischrechner/% |
| 121PD | AL | | | 29 | X | | 1 | 12 | X | X | | | X | DIG | 12 + 1 | X | X | X | Drucker | kleiner Tischrechner/%/$^+$/_ /M+/M– |

| Hersteller Typ | Rechenlogik | Anzahl der Klammern | Anzahl der Funktionen | Anzahl der Tasten | Tastenbeleg. einfach | Tastenbeleg. mehrfach | Zahl der Speicher | Zahl der Stellen | Komma Fest- | Komma Fließ- | Komma Gleit- | techn.-wiss. | kaufmännisch | Anzeige Art | Anzeige Stellen | Stromversorgung Akku | Stromversorgung Batterie | Stromversorgung Netz | Ergänzungen | Besonderheiten |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| noch | | | | | | | | | | | | | | | | | | | | |
| **NASCO** | | | | | | | | | | | | | | | | | | | | |
| 1221P | ARL | | 1 | 32 | X | | 2 | 12 | X | X | | | X | | | | | X | Drucker (12 Stellen) | Tischrechner/%/+/_/M+/M– |
| 1221PD | ARL | | 1 | 32 | X | | 2 | 12 | X | X | | | X | DIG | 12 + 1 | | | X | wie 1221P | wie 1221P |
| **National** | | | | | | | | | | | | | | | | | | | | |
| E-38-BK | AL | | | 22 | X | | 1 | 8 | | X | | | | LED | 8 + 1 | | X | | | % |
| N-36-BK | UPN | | | 15 | X | | | | X | | | | | LED | 6 + 1 | | X | | | |
| NS4640 | UPN | | 24 | 36 | | X | 3 | 10 | X | X | X | X | | LED | 12 + 2 | X | | X | | π/+/_/%/Δ%/M+/rad/Statistik/Umrechnungen |
| Novus 650 | UPN | | | 16 | X | | | | X | | | | | LED | 6 + 1 | | X | X | | |
| Novus 750 | AL | | | 17 | X | | | | | X | | | | LED | 6 + 1 | | X | X | | |
| Novus 822 | AL | | | 20 | X | | | | | X | | | | LED | 8 + 1 | | X | X | | % |
| Novus 824 | AL | | | 23 | X | | 1 | 8 | | X | | | | LED | 8 + 1 | | X | X | | %/+/_/M+/M– |
| Novus 826 | AL | | | 22 | X | | 1 | 8 | | X | | | | LED | 8 + 1 | | X | X | | % |
| Novus 831 | AL | | 2 | 20 | X | | | | | X | | | | LED | 8 + 1 | | X | X | | % |
| Novus 832 | AL | | 2 | 22 | X | | | | | X | | | | LED | 8 + 1 | | X | X | | %/+/_ |
| Novus 835 | AL | | 1 | 23 | X | | 1 | 8 | | X | | | | LED | 8 + 1 | | X | X | | %/M+/M– |
| Novus 842 | AL | | 1 | 23 | X | | 1 | 8 | | X | | | | LED | 8 + 1 | | X | X | | %/M+/M– |
| Novus 4510 | UPN | | 13 | 32 | | X | 1 | 8 | | X | | X | | LED | 8 + 1 | | X | X | | π/+/_/M+/M–/M + $x^2$/rad |
| Novus 4520 | UPN | | 13 | 35 | X | | 1 | 8 | | X | X | X | | LED | 10 + 2 | | X | X | | π/+/_ |
| Novus 6010 | ARL | | 2 | 32 | | X | 1 | 8 | | X | | | | LED | 8 + 1 | | X | X | | π/%/+/_/M+/Umrechnungen |
| Novus 6020 | ARL | | 10 | 32 | | X | 1 | 8 | | X | | | X | LED | 8 + 1 | | X | X | | +/_/%/Δ%/M+/Finanztechnik |
| Novus 6030 | ARL | | 10 | 32 | X | | 1 | 8 | | X | | X | | LED | 8 + 1 | | X | X | | %/+/_/M+/Statistikrechner |
| **NIKKO** | | | | | | | | | | | | | | | | | | | | |
| 10P | ARL | | | 26 | X | | 1 | 10 | X | X | | | X | | | | | X | Drucker (10 Stellen) | Tischrechner/%/M+/M– |
| 10PD | ARL | | | 26 | X | | 1 | 10 | X | X | | | X | DIG | 10 + 1 | | | X | wie 10P | wie 10P |
| 120MPD | ARL | | | 33 | X | | 1 | 12 | X | X | | | X | DIG | 12 + 1 | | | X | Drucker | Tischrechner/%/Δ%/M+/M– |
| 810MP | ARL | | | 28 | X | | 1 | 10 | X | X | | | X | | | | | X | Drucker (10 Stellen) | Tischrechner/%/+/_/M+/M– |
| 810MPD | ARL | | | 28 | X | | 1 | 10 | X | X | | | X | DIG | 10 + 1 | | | X | wie 810MP | wie 810MP |
| 812MP | ARL | | | 28 | X | | 1 | 12 | X | X | | | X | | | | | X | Drucker (12 Stellen) | wie 810 |
| 812MPD | ARL | | | 28 | X | | 1 | 12 | X | X | | | X | DIG | 12 + 1 | | | X | wie 812MP | wie 812MP |
| 812P | ARL | | | 28 | X | | | | X | X | | | X | | | | | X | Drucker (12 Stellen) | Tischrechner/%/+/_ |
| 812PD | ARL | | | 28 | X | | | | X | X | | | X | DIG | 12 + 1 | | | X | wie 812P | wie 812P |
| 912MPD | AL | | | 29 | X | | 1 | 12 | X | X | | | X | DIG | 12 + 1 | X | X | X | Drucker | kleiner Tischrechner/%/+/_/M+/M– |
| **Nissin** | | | | | | | | | | | | | | | | | | | | |
| LC-850M | AL | 1 | 2 | 30 | X | | 1 | 8 | | X | | | | LCD | 8 + 1 | | X | | | π/%/+/_/M+/M– |
| LCD-009 | AL | | 1 | 24 | X | | 1 | 8 | | X | | | | LCD | 8 + 1 | | X | | | %/+/_/M+/M– |
| LCD-012 | AL | | 1 | 24 | X | | 1 | 8 | | X | | | | LCD | 8 + 1 | | X | | | wie 009 |
| MP-101 | AL | | | 31 | X | | 1 | 10 | X | X | | | X | DIG | 10 + 1 | X | | X | Drucker | Tischrechner/%/+/_/M+/M– |
| MPR-400 | AL | | | 31 | X | | 1 | 10 | X | X | | | X | DIG | 10 + 1 | X | | X | wie 101 | wie 101 |
| SLC-3000A | AL | 2 | 20 | 35 | | X | 1 | 8 | | | X | X | | LCD | 10 + 2 | | X | | | π/+/_/M+/Statistik |
| SLC-6000B | AL | 2 | 16 | 35 | | X | 1 | 8 | | | X | X | | LCD | 7 + 2 | | X | | | π/+/_/M+ |
| **Olivetti** | | | | | | | | | | | | | | | | | | | | |
| Logos 3 | ARL | | | 24 | X | | 1 | 12 | X | X | | | X | | | X | X | X | Drucker (12 Stellen) | Taschenrechnerformat |
| Logos 7 | ARL | | | 27 | X | | 1 | 12 | X | X | | | X | | 12 + 1 | X | X | X | Drucker | wie Logos 3, zusätzlich Anzeige/Δ%/Kalender |
| Logos 245 | ARL | | | 25 | X | | 2 | 16 | X | X | | | X | | | | | X | Drucker (16 Stellen) | Tischrechner/Δ% |
| **Olympia** | | | | | | | | | | | | | | | | | | | | |
| CA12 | ARL | | | 24 | X | | | | X | X | | | X | | | | | X | Drucker (12 Stellen) | Tischrechner/%/durch 100/durch 1000 |

| Hersteller Typ | Rechenlogik | Anzahl der Klammern | Anzahl der Funktionen | Anzahl der Tasten | Tastenbeleg. einfach | Tastenbeleg. mehrfach | Zahl der Speicher | Zahl der Stellen | Komma Fest- | Komma Fließ- | Komma Gleit- | techn.-wiss. | kaufmännisch | Anzeige Art | Anzeige Stellen | Stromversorgung Akku | Stromversorgung Batterie | Stromversorgung Netz | Ergänzungen | Besonderheiten |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| noch | | | | | | | | | | | | | | | | | | | | |
| ● **Olympia** | | | | | | | | | | | | | | | | | | | | |
| CA120 | ARL | | | 24 | X | | | | X | X | | | X | | | | | X | Drucker (12 Stellen) | wie CA12 |
| CD41 | AL | | | 19 | X | | | | | X | | | | DIG | 8 + 1 | | X | X | | % |
| CD42 | AL | | 1 | 22 | X | | 1 | 8 | | X | | | | DIG | 8 + 1 | | X | X | | %/+/– |
| CD45 | AL | | 1 | 20 | X | | | | | X | | | | DIG | 8 + 1 | | X | X | | % |
| CD46 | AL | | 1 | 24 | X | | 1 | 8 | | X | | | | DIG | 8 + 1 | | X | X | | %/+/–/M+/M– |
| CD47 | AL | | 3 | 28 | X | | 1 | 8 | | X | | | | DIG | 8 + 1 | | X | X | | π/%/+/–/M+/M– |
| CD71 | ARL | | | 21 | X | | 1 | 8 | X | X | | | | DIG | 8 + 1 | X | | X | | %/+/– |
| CD72 | AL | | 1 | 21 | X | | 1 | 8 | X | X | | | | DIG | 8 + 1 | X | | X | | π/+/– |
| CD75 | ARL | | | 21 | X | | 1 | 8 | X | X | | | X | DIG | 8 + 1 | X | | X | | %/+/– |
| CD76 | AL | | 1 | 23 | X | | 1 | 8 | | X | | | | DIG | 8 + 1 | X | | X | | %/+/–/M+ |
| CD91 | AL | | 12 | 34 | X | | 1 | 8 | | X | X | X | | DIG | 7 + 2 | | X | X | | π/+/–/M+/rad |
| CD92 | AL | 2 | 15 | 34 | | X | 1 | 8 | | X | X | X | | DIG | 10 + 2 | | X | X | | wie CD91, zusätzlich Statistik |
| CD101 | ARL | | | 19 | X | | | | X | X | | | X | DIG | 10 + 1 | | X | X | | Tischrechner/% |
| CD102 | ARL | | | 20 | X | | | | X | X | | | X | DIG | 12 + 1 | | X | X | | Tischrechner/% |
| CD402 | ARL | | | 26 | X | | 1 | 12 | X | X | | | X | DIG | 12 + 1 | | | X | | Tischrechner/%/+/–/M+/M– |
| CD502 | ARL | | 1 | 30 | X | | 2 | 14 | X | X | | | X | DIG | 14 + 1 | | | X | | Tischrechner/wie CD402 |
| CD602 | AL | | 12 | 26 | | X | 1 | 8 | | X | | X | | DIG | 8 + 1 | | | X | | Tischrechner/rad |
| CD603 | AL | | 20 | 35 | | X | 1 | 10 | | X | X | X | | DIG | 12 + 2 | | | X | | Tischrechner/π/+/–/rad/Hyp./M+ |
| CP140 | ARL | | | 28 | X | | | | X | X | | | X | | | | | X | Drucker (12 Stellen) | Tischrechner/% |
| CP141 | ARL | | | 28 | X | | 1 | 12 | X | X | | | X | | | | | X | wie CP140 | wie CP140 |
| CP162 | ARL | | 1 | 33 | X | | 2 | 14 | X | X | | | X | | | | | X | Drucker (14 Stellen) | Tischrechner/%/+/– |
| CP181 | ARL | | 1 | 30 | X | | 1 | 16 | X | X | | | X | | | | | | Drucker (16 Stellen) | Tischrechner/%/+/– |
| CP1202 | ARL | | | 26 | X | | | | X | X | | | X | | | | | | wie CP181 | wie CP181 |
| CP1210 | ARL | | | 30 | X | | 1 | 12 | X | X | | | X | | | | | X | Drucker (12 Stellen) | Tischrechner |
| CP1211 | ARL | | | 31 | X | | 1 | 12 | X | X | | | X | | | | | X | wie CP1210 | Tischrechner/Δ% |
| CPA1200 | ARL | | | 24 | X | | 1 | 12 | X | X | | | X | | | | | X | wie CA12 | wie CA12 |
| CPD5210 | ARL | | | 30 | X | | 1 | 12 | X | X | | | X | DIG | 12 + 1 | | | X | Drucker | Tischrechner |
| CPD5211 | ARL | | | 31 | X | | 1 | 12 | X | X | | | X | DIG | 12 + 1 | | | X | wie CPD5210 | Tischrechner/Δ% |
| CPD5212 | ARL | | | 29 | X | | 1 | 12 | X | X | | | X | DIG | 12 + 1 | | | X | wie CPD5211 | wie CPD5211/M+/M– |
| CPD5420 | ARL | | | 34 | X | | 2 | 14 | X | X | | | X | DIG | 14 + 1 | | | X | Drucker | Tischrechner/Δ% |
| CPK120 | ARL | | | 22 | X | | | | X | X | | | X | | | | | X | Drucker (12 Stellen) | Tischrechner |
| CPM12 | ARL | | | 22 | X | | | | X | X | | | X | | | X | | X | Drucker (12 Stellen) | Tischrechner/% |
| CPM12/1 | ARL | | | 22 | X | | 1 | 12 | X | X | | | X | | | X | | X | wie CPM12 | Tischrechner/% |
| HIT2000 | AL | | 1 | 27 | X | | 1 | 8 | | X | | | | LCD | 8 + 1 | | X | | | %/+/–/M+/M–/getrennte Zeitanzeige/Wecker/1600 Stunden |
| LCD1600 | AL | | 1 | 24 | X | | 1 | 8 | | X | | | | LCD | 8 + 1 | | X | | | %/+/–/M+/M–/1600 Stunden |
| LCD1800 | AL | | 1 | 24 | X | | 1 | 8 | | X | | | | LCD | 8 + 1 | | X | | | wie LCD1600 |
| ● **Omron** | | | | | | | | | | | | | | | | | | | | |
| 86 | AL | | | 20 | X | | | | | X | | | | DIG | 8 + 1 | X | X | X | | %/+/– |
| 86M | AL | | | 20 | X | | 1 | 8 | X | X | | | | DIG | 8 + 1 | X | X | X | | %/+/–/M+/M– |
| 86R | AL | | 1 | 24 | X | | | | X | X | | | | DIG | 8 + 1 | X | X | X | | % |
| 86SR | AL | | 10 | 19 | | X | 1 | 8 | | X | X | X | | DIG | 7 + 2 | X | X | X | | π/M+/M–/rad |
| ● **PRECISA** | | | | | | | | | | | | | | | | | | | | |
| 512 | ARL | | 1 | 24 | X | | 1 | 12 | X | X | | | X | DIG | 12 + 1 | | | X | | Tischrechner/%/+/–/M+/M– |
| 2300 | ARL | | | 32 | X | | 1 | 14 | X | X | | | X | DIG | 14 + 1 | | | X | Drucker | Tischrechner/%/+/–/M+/M– |
| 2400 | ARL | | | 37 | X | | 2 | 14 | X | X | | | X | DIG | 14 + 1 | | | X | wie 2300 | wie 2300 |
| 3101 | ARL | | | 25 | X | | | | X | X | | | X | | | | | X | Drucker (12 Stellen) | Tischrechner/%/+/–/durch 100 |
| 3201 | ARL | | | 26 | X | | | | X | X | | | X | | | | | | wie 3101 | wie 3101 |

| Hersteller Typ | Rechenlogik | Anzahl der Klammern | Anzahl der Funktionen | Anzahl der Tasten | Tastenbeleg. einfach | Tastenbeleg. mehrfach | Zahl der Speicher | Zahl der Stellen | Komma Fest- | Komma Fließ- | Komma Gleit- | techn.-wiss. | kaufmännisch | Anzeige Art | Anzeige Stellen | Stromversorgung Akku | Stromversorgung Batterie | Stromversorgung Netz | Ergänzungen | Besonderheiten |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| noch | | | | | | | | | | | | | | | | | | | | |
| • **PRECISA** | | | | | | | | | | | | | | | | | | | | |
| 3401 | ARL | | | 29 | X | | 1 | 12 | X | X | | | X | DIG | 12 + 1 | | | X | Drucker | Tischrechner/%/+/_ /M+/M– |
| • **Quelle-Privileg** | | | | | | | | | | | | | | | | | | | | |
| 8 | ARL | | | 18 | X | | | | | X | | | | DIG | 8 + 1 | | | X | | Tischrechner |
| 12PAM | ARL | | | 22 | X | | | | X | X | | | X | D | | | | X | Drucker (12 Stellen) | Tischrechner/% |
| 12PPS | ARL | | | 18 | X | | | | X | | | | | | | | | X | Drucker (6 Stellen) | Tischrechner |
| 16PPMD | ARL | | | 22 | X | | 1 | 8 | X | | | | X | | | | | X | Drucker (8 Stellen) | Tischrechner |
| 69D-E | AL | | 2 | 27 | X | | 1 | 8 | | X | X | | | DIG | 8 + 1 | | X | X | | π/M+/M– |
| 81 | AL | | | 18 | X | | | | | X | | | | LED | 8 + 1 | | X | X | | |
| 82 | AL | | | 17 | X | | | | | X | | | | LED | 8 + 1 | | X | X | | |
| 85 | AL | | | 18 | X | | | | | X | | | | LED | 8 + 1 | | X | X | | % |
| 583D-E | AL | | 13 | 38 | X | | 1 | 8 | | X | X | X | | DIG | 7 + 2 | | X | X | | π/M+/M–/rad |
| 685D-E-NC | AL | 2 | 14 | 36 | X | | 1 | 8 | | X | X | X | | DIG | 7 + 2 | X | | X | | π/+/_ /rad |
| 800D | ARL | | | 18 | X | | | | | X | | | | LED | 8 + 1 | | X | X | | Tischrechner/% |
| 800K | ARL | | | 18 | X | | | | X | | | | | LED | 8 + 1 | | X | X | | Tischrechner |
| 800M | AL | | | 21 | X | | 1 | 8 | | X | | | | LED | 8 + 1 | | X | X | | Tischrechner/%/+/_ |
| 801 | ARL | | 1 | 19 | X | | | | | X | | | | DIG | 8 + 1 | | | X | | % |
| 800R | ARL | | 1 | 20 | X | | | | | X | | | | LED | 8 + 1 | | X | X | | Tischrechner/%/+/_ |
| 803 | AL | | | 19 | X | | | | | X | | | | LED | 8 + 1 | | X | X | | % |
| 804D | AL | | | 18 | X | | | | | X | | | | DIG | 8 + 1 | | X | X | | % |
| 807D | AL | | | 18 | X | | | | | X | | | | DIG | 8 + 1 | | X | X | | % |
| 808D | AL | | | 19 | X | | | | | X | | | | DIG | 8 + 1 | | X | X | | %/+/_ |
| 810D-NC | AL | | | 18 | X | | | | | X | | | | DIG | 8 + 1 | X | | X | | % |
| 811M | ARL | | 1 | 23 | X | | 1 | 8 | | X | | | | DIG | 8 + 1 | | | X | | %/+/_ /M+/M– |
| 815D | AL | | | 20 | X | | | | | X | | | | DIG | 8 + 1 | | X | X | | % |
| 830D | AL | | 1 | 20 | X | | | | | X | | | | DIG | 8 + 1 | | X | X | | % |
| 831NC | AL | | 3 | 25 | X | | | | | X | X | | | LED | 7 + 2 | X | | X | | π/%/+/_ |
| 832D | AL | | 1 | 21 | X | | | | | X | | | | DIG | 8 + 1 | | X | X | | π/% |
| 833K | AL | | | 19 | X | | 1 | 8 | | X | | | | LED | 8 + 1 | | X | X | | % |
| 835M | AL | | | 24 | X | | 1 | 8 | | X | | | | LED | 8 + 1 | | X | X | | %/+/_ /M+/M– |
| 840M | AL | | 2 | 26 | X | | 1 | 8 | | X | | | | LED | 8 + 1 | | X | X | | %/+/_ /M+/M– |
| 852M-NC | ARL | | 1 | 25 | X | | 1 | 8 | | X | | | | LED | 8 + 1 | | X | X | | %/+/_ |
| 856MD | AL | | 1 | 27 | X | | 1 | 8 | | X | | | | DIG | 8 + 1 | | X | X | | π/%/+/_ /M+/M– |
| 860MD | AL | | 1 | 23 | X | | 1 | 8 | | X | | | | DIG | 8 + 1 | | X | X | | %/M+/M– |
| 860MD-NC | AL | | 1 | 23 | X | | 1 | 8 | | X | | | | DIG | 8 + 1 | X | | X | | %/M+/M– |
| 863M | AL | | 2 | 24 | X | | 1 | 8 | | X | | | | LED | 8 + 1 | | X | X | | π/%/+/_ |
| 864MD | AL | | 3 | 24 | X | | 1 | 8 | | X | | | | DIG | 8 + 1 | | X | X | | %/+/_ |
| 867MD | AL | | 3 | 25 | X | | 1 | 8 | | X | | | | DIG | 8 + 1 | | X | X | | %/M+/M– |
| 868MD | AL | | 1 | 22 | X | | 1 | 8 | | X | | | | DIG | 8 + 1 | | X | X | | %/+/_ |
| 869SF | AL | | 1 | 22 | X | | 1 | 8 | | X | | | | DIG | 8 + 1 | | X | | | %/+/_ |
| 870ESR-E | AL | | 6 | 29 | X | | 1 | 8 | | X | X | X | | LED | 10 + 2 | | X | X | | +/_ |
| 871ES-NC | AL | 1 | 2 | 25 | X | | | | | X | X | | | LED | 8 + 2 | X | | X | | π/+/_ |
| 872MD | AL | 1 | 3 | 30 | X | | 1 | 8 | | X | | | | DIG | 8 + 1 | | X | X | | %/+/_ /M+/M– |
| 875MD-NC | AL | 1 | 3 | 30 | X | | 1 | 8 | | X | | | | DIG | 8 + 1 | X | | X | | wie 872MD |
| 878MD | AL | | 1 | 22 | X | | 1 | 8 | | X | | | | DIG | 8 + 1 | | X | X | | %/+/_ |
| 883D-ESR-E | AL | | 10 | 35 | X | | 1 | 8 | | X | X | X | | DIG | 7 + 2 | | X | X | | π/M+/M– |
| 890FIR-D | AL | | 15 | 20 | | X | 4 | 8 | | X | | | X | DIG | 8 + 1 | | X | X | | %/Δ %/+/_ /Finanzrechner |
| 891STR-NC | ARL | | 8 | 25 | X | | 3 | 8 | | X | | | X | LED | 8 + 1 | X | | X | | Statistikrechner |
| 1080ESR-E-NC | AL | 2 | 13 | 39 | X | | 1 | 10 | | | X | X | | LED | 12 + 2 | X | | X | | π/+/_ /rad |
| 1081ESR-E | AL | | 17 | 35 | X | | 1 | 10 | | | X | X | | LED | 12 + 2 | | X | X | | π/+/_ /M+/rad |
| 1200 | AL | | | 19 | X | | | | X | X | | | | LED | 12 + 2 | | | X | | Tischrechner |
| 1200-Memory | ARL | | | 23 | X | | 1 | 12 | X | X | | | X | | | | | X | Drucker (12 Stellen) | Tischrechner/%/M+/M– |
| 1200-Printer | ARL | | | 19 | X | | | | X | X | | | X | | | | | X | wie 1200-Memory | Tischrechner/% |

| Hersteller Typ | Rechenlogik | Anzahl der Klammern | Anzahl der Funktionen | Anzahl der Tasten | Tastenbeleg. einfach | Tastenbeleg. mehrfach | Zahl der Speicher | Zahl der Stellen | Komma Fest- | Komma Fließ- | Komma Gleit- | techn.-wiss. | kaufmännisch | Anzeige Art | Anzeige Stellen | Stromversorgung Akku | Stromversorgung Batterie | Stromversorgung Netz | Ergänzungen | Besonderheiten |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| noch | | | | | | | | | | | | | | | | | | | | |
| **• Quelle-Privileg** | | | | | | | | | | | | | | | | | | | | |
| 1201M | AL | | | 23 | X | | 1 | 12 | X | X | | | | LED | 12 + 2 | | | X | | Tischrechner/M+/M– |
| 1202PM | ARL | | 1 | 31 | X | | 1 | 12 | X | X | | | X | | | | | X | Drucker (12 Stellen) | Tischrechner/%/M+/M– |
| 1212PMD | ARL | | | 31 | X | | 1 | 12 | X | X | | | X | DIG | 12 + 1 | | | X | wie 1202PM | wie 1202PM |
| 1500LC | AL | | 1 | 20 | X | | | | | X | | | | LCD | 8 + 1 | | X | | | %/1500 Stunden |
| Calculator | AL | | | 18 | X | | | | | X | | | | LED | 8 + 1 | | X | X | | |
| CX-800 | ARL | | | 17 | X | | | | | X | | | | DIG | 8 + 1 | | | X | | Tischrechner |
| CX-1200M | ARL | | | 21 | X | | 1 | 12 | | X | | | | DIG | 12 + 1 | | | X | | Tischrechner/+/– /M+ |
| ESR | AL | | 12 | 25 | | X | 1 | 8 | | X | | | | DIG | 8 + 1 | | X | X | | π/+/– /M+ |
| F-892NC | AL | | 18 | 46 | X | | 2 | 10 | | X | X | | X | LED | 12 + 2 | X | | X | | %/+/– /Δ %/M+/Finanzrechner |
| LC10000 | AL | | 1 | 30 | X | | 1 | 8 | | X | | | | LCD | 8 + 1 | | X | | | %/M+/M–/Kalender/Stoppuhr/Wecker/ 10 000 Stunden |
| LCD2000M | AL | | 1 | 23 | X | | 1 | 8 | | X | | | | LCD | 8 + 1 | | X | | | %/+/– /M+/M–/1500 Stunden |
| Micro-Computer | ARL | | | 16 | X | | | | X | X | | | | LED | 8 + 1 | | X | | | |
| Mini-Calculator | ARL | | | 19 | X | | | | X | X | | | | LED | 8 + 1 | | X | X | | % |
| Mini-Memory | ARL | | | 20 | X | | 1 | 8 | | X | | | | LED | 8 + 1 | | X | X | | M+/M– |
| Mini-Computer | ARL | | | 16 | X | | | | | X | | | | LCD | 8 + 1 | | X | | | |
| NB800 | ARL | | | 16 | X | | | | | X | | | | DIG | 8 + 1 | | X | X | | |
| Portable-low-noise | ARL | | | 22 | X | | | | X | X | | | X | | | X | | X | Drucker (12 Stellen) | Tischrechner/% |
| SR-30 | AL | 1 | 16 | 39 | | X | 2 | 8 | | X | X | X | | LED | 10 + 2 | | X | X | | π/+/– /M+/rad/Statistik |
| SR-35NC | AL | 15 | 15 | 40 | X | | 1 | 11 | | X | X | X | | LED | 8 + 1 | X | | X | | π/%/+/– /M+ |
| SR-54NC | UPN | | 14 | 30 | | X | 9 | 10 | X | X | X | X | | LED | 12 + 2 | X | | X | | π/%/Δ %/+/– /M+/rad/Polar/Statistik |
| SR-58D-NC | AL | 2 | 14 | 35 | | X | 1 | 8 | X | X | X | X | | DIG | 10 + 2 | X | | X | | π/+/– /M+/M–/M x/M ÷/rad |
| SR-60D-NC | AL | 5 | 14 | 35 | | X | 10 | 8 | X | X | X | X | | DIG | 10 + 2 | X | | X | | wie SR-58D-NC |
| SR-69D-E | AL | | 1 | | | | 1 | 8 | | X | X | X | | DIG | 7 + 2 | | X | X | | π/M+ |
| SR-80D-NC | AL | 1 | 18 | 39 | | X | 2 | 8 | | X | X | X | | DIG | 10 + 2 | X | | X | | π/+/– /M+/rad/Polar/Statistik |
| SR-100NC | AL | 1 | ca. 35 | 49 | | X | 2 | 10 | | | X | X | | LED | 12 + 2 | X | | X | | π/+/– /M+/rad/Polar/Statistik/komplexe Rechnung/Umrechnungen |
| **• Radofin** | | | | | | | | | | | | | | | | | | | | |
| 8P | AL | | | 19 | X | | | | | X | | | | DIG | 8 + 1 | | X | | | % |
| 1620 | AL | | | 19 | X | | 1 | 8 | | X | | | | DIG | 8 + 1 | | X | | | %/M+/M– |
| 2560 | AL | | 10 | 20 | | X | 1 | 8 | | X | X | X | | DIG | 7 + 2 | | X | | | π/+/– /M+/M– |
| **• Realtone** | | | | | | | | | | | | | | | | | | | | |
| 8413 | ARL | | 2 | 20 | X | | | | | X | | | | DIG | 8 + 1 | | X | X | | +/– |
| KP200 | AL | | 3 | 24 | X | | 1 | 8 | | X | | | | DIG | 8 + 1 | | X | X | | % |
| KP300 | ARL | | | 22 | X | | 1 | 8 | | X | | | | DIG | 8 + 1 | X | | X | | M+/M– |
| MC60 | AL | | | 20 | X | | 1 | 8 | | X | | | | DIG | 8 + 1 | X | | X | | %/Umrechnungen |
| SC-40 | AL | 2 | 13 | 39 | X | | 1 | 10 | | X | X | X | | DIG | 12 + 2 | X | | X | | π/+/– /rad |
| **• Reno** | | | | | | | | | | | | | | | | | | | | |
| 941 | AL | | 1 | 25 | X | | 1 | 8 | | X | | | | DIG | 8 + 1 | | X | | | %/+/– /M+/M– |
| 941R | AL | | 1 | 25 | X | | 1 | 8 | | X | | | | DIG | 8 + 1 | X | | X | | wie 941 |
| 952 | AL | 2 | 13 | 36 | X | | 1 | 8 | | X | X | X | | DIG | 7 + 2 | | X | | | π/%/+/– /rad |
| 952R | AL | 2 | 13 | 36 | X | | 1 | 8 | | X | X | X | | DIG | 7 + 2 | X | | X | | wie 952 |
| 1382 | AL | 2 | 14 | 37 | | X | 1 | 10 | | X | X | X | | DIG | 10 + 2 | | X | | | π/+/– /rad/Polar/M+ |
| 1382R | AL | 2 | 14 | 37 | | X | 1 | 10 | | X | X | X | | DIG | 10 + 2 | X | | X | | wie 1382 |
| **• Rockwell** | | | | | | | | | | | | | | | | | | | | |
| 8R | AL | | | 18 | X | | | | | X | | | | LED | 8 + 1 | | X | X | | % |
| 10R | ARL | | | 16 | X | | | | | X | | | | LED | 8 + 1 | | X | X | | |
| 18R | AL | | | 20 | X | | 1 | 8 | | X | | | | LED | 8 + 1 | | X | X | | % |
| 18T | AL | | | 20 | X | | 1 | 8 | | X | | | | DIG | 8 + 1 | | X | X | | % |
| 20R | AL | | | 20 | X | | 1 | 8 | | X | | | | LED | 8 + 1 | | X | X | | %/M+ |
| 20T | AL | | | 23 | X | | 1 | 8 | | X | | | | DIG | 8 + 1 | X | | X | | %/+/– /M+/M– |

| Hersteller Typ | Rechenlogik | Anzahl der Klammern | Anzahl der Funktionen | Anzahl der Tasten | Tastenbeleg. einfach | Tastenbeleg. mehrfach | Zahl der Speicher | Zahl der Stellen | Komma Fest- | Komma Fließ- | Komma Gleit- | techn.-wiss. | kaufmännisch | Anzeige Art | Anzeige Stellen | Stromversorgung Akku | Stromversorgung Batterie | Stromversorgung Netz | Ergänzungen | Besonderheiten |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| noch | | | | | | | | | | | | | | | | | | | | |
| ● **Rockwell** | | | | | | | | | | | | | | | | | | | | |
| 21R | AL | | | 20 | X | | 1 | 8 | | X | | | | LED | 8 + 1 | X | | X | | wie 20R |
| 22K | AL | | 1 | 24 | X | | 1 | 8 | | X | | | | DIG | 8 + 1 | | X | X | | %/+/_/M+/M– |
| 24K | AL | | 1 | 25 | X | | 1 | 8 | | X | | | | DIG | 8 + 1 | X | | X | | %/+/_/M+/M– |
| 24RD | AL | | 1 | 24 | X | | 1 | 8 | | X | | | | DIG | 8 + 1 | X | X | X | | %/+/_/M+/M– |
| 30R | AL | | 3 | 20 | | X | 1 | 8 | | X | | | | LED | 8 + 1 | | X | X | | %/+/_/M+/M– |
| 31R | AL | | 3 | 20 | | X | 1 | 8 | | X | | | | LED | 8 + 1 | X | | X | | wie 30R |
| 44RD | AL | 2 | 14 | 36 | X | | 1 | 8 | | X | X | X | | DIG | 7 + 2 | X | X | X | | π/+/_ |
| 51R | AL | | | 20 | | X | 2 | 8 | X | X | | | | DIG | 8 + 1 | X | | X | | M+/M–/Bruchrechnungen/Umrechnungen |
| 61R | AL | | 10 | 20 | | X | 1 | 8 | | X | | X | | DIG | 8 + 1 | X | | X | | π/M+/M–/M + $x^2$/+/_ |
| 63R | AL | 2 | 14 | 25 | | X | 1 | 8 | | X | X | X | | DIG | 10 + 2 | X | | X | | π/+/_/M+/M–/M x/M ÷/rad |
| 64RD | AL | 2 | 14 | 37 | | X | 1 | 8 | | X | X | X | | DIG | 10 + 2 | X | | X | | π/+/_/M+/rad |
| 74K | AL | 2 | 14 | 25 | | X | 1 | 8 | | X | X | X | | DIG | 10 + 2 | X | | X | | π/+/_/M+/M–/M x/M ÷/rad |
| 202 | AL | | 14 | 20 | | X | 1 | 8 | | X | | X | | DIG | 8 + 1 | X | X | X | | π/+/_/M+/M–/M + $x^2$/rad |
| 203 | AL | | | 20 | | X | 2 | 8 | X | X | | | | DIG | 8 + 1 | X | X | X | | wie 51R |
| 204 | AL | | | 20 | | X | 1 | 8 | | X | | | X | DIG | 8 + 1 | X | X | X | | Finanzrechner |
| 212P | ARL | | | 22 | X | | | | | X | | | X | | | | | X | Drucker (12 Stellen) | Tischrechner/% |
| 222P | ARL | | | 24 | X | | | | X | X | | | X | | | | | X | Drucker (12 Stellen) | Tischrechner/% |
| 232P | ARL | | 1 | 29 | X | | 1 | 12 | X | X | | | X | | | | | X | Drucker (12 Stellen) | Tischrechner/%/M+/M– |
| 310 | ARL | | | 21 | X | | 1 | 8 | X | X | | | X | DIG | 8 + 1 | | | X | | Tischrechner/%/+/_/M+ |
| 320 | ARL | | | 26 | X | | 1 | 12 | X | X | | | X | DIG | 12 + 1 | | | X | | %/M+/M– |
| 330 | ARL | | 2 | 30 | X | | 2 | 14 | X | X | | | | DIG | 14 + 1 | | | X | | %/M+/M– |
| 350 | AL | 2 | 13 | 36 | X | | | | | X | X | X | | DIG | 10 + 2 | | | X | | π/+/_/M+/M– |
| 425P | ARL | | | 24 | X | | 1 | 12 | X | X | | | X | | | | | X | Drucker (12 Stellen) | Tischrechner/% |
| 435P | ARL | | | 28 | X | | 1 | 12 | X | X | | | X | | | | | X | Drucker (12 Stellen) | Tischrechner/%/M+/M– |
| 445P/D | ARL | | | 30 | X | | 1 | 12 | X | X | | | X | DIG | 12 + 1 | | | X | Drucker | Tischrechner/%/M+/M– |
| 470P/D | ARL | | | 33 | X | | 1 | 12 | X | X | | | X | DIG | 12 + 1 | | | X | Drucker | Tischrechner/%/M+/M–/durch 100 |
| Anita 810 | AL | | | 18 | X | | | | X | X | | | | LED | 8 + 1 | | X | X | | % |
| Anita 811 | AL | | | 20 | X | | 1 | 8 | X | X | | | | LED | 8 + 1 | | X | X | | % |
| Anita 831 | AL | | 1 | 20 | X | | 1 | 8 | X | X | | | | LED | 8 + 1 | | X | X | | %/M+ |
| Anita 841 | AL | | 13 | 20 | | X | 1 | 8 | | X | | X | | LED | 8 + 1 | | X | X | | π/+/_/M+/M–/M + $x^2$/rad |
| Anita 851 | AL | | | 20 | | X | 2 | 8 | X | X | | | | LED | 8 + 1 | | X | X | | wie 51R und 203 |
| Anita 861 | AL | | | 20 | | X | 1 | 8 | | X | | | X | LED | 8 + 1 | | X | X | | wie 204 |
| Blackjack | AL | | | 25 | X | | | | | X | | | | DIG | 8 + 1 | | X | | | Rechner für Spiele |
| ● **ROYAL** | | | | | | | | | | | | | | | | | | | | |
| 100P | ARL | | | 24 | X | | | | X | X | | | X | | | | | X | Drucker (12 Stellen) | kleiner Tischrechner/% |
| 300P | ARL | | | 25 | X | | | | X | X | | | X | | | | | X | wie 100P | wie 100P |
| 300PD | ARL | | | 25 | X | | | | X | X | | | X | DIG | 12 + 1 | | | X | wie 300P | wie 300P |
| 410P | ARL | | | 29 | X | | 1 | 12 | X | X | | | X | | | | | X | wie 300P | wie 300P/M+/M– |
| 450PD | ARL | | 1 | 32 | X | | 2 | 12 | X | X | | | X | DIG | 12 + 1 | | | X | wie 410P | wie 410P |
| ● **Sanyo** | | | | | | | | | | | | | | | | | | | | |
| 2051P | ARL | | 1 | 30 | X | | 1 | 10 | X | X | | | X | | | | | X | Drucker | Tischrechner/%/M+/M– |
| 2161P | ARL | | 1 | 30 | X | | 1 | 10 | X | X | | | X | | | | | X | wie 2051P | wie 2051P |
| CX-1251 | AL | | | | | | 1 | 8 | | X | | | | LCD | 8 + 1 | | X | | | |
| CX-2005 | AL | | | 20 | X | | | | | X | | | | DIG | 6 + 1 | | X | X | | % |
| CX-7250H | AL | | 1 | 28 | X | | 1 | 8 | | X | | | | LCD | 8 + 1 | | X | | | %/M+/M–/Schrittzähler/8 000 Stunden |
| CX-8014 | AL | | 1 | 22 | X | | | | | X | | | | DIG | 8 + 1 | | X | X | | π/%/+/_ |
| CX-8016 | AL | | | 20 | X | | | | | X | | | | DIG | 8 + 1 | | X | X | | %/+/_ |
| CX-8027L | AL | | 1 | 21 | X | | | | | X | | | | LCD | 8 + 1 | | X | | | %/+/_/1500 Stunden |
| CX-8105 | AL | | 1 | 22 | X | | 1 | 8 | | X | | | | DIG | 8 + 1 | | X | X | | %/+/_ |
| CX-8179L | AL | | 1 | 23 | X | | 1 | 8 | | X | | | | LCD | 8 + 1 | | X | | | %/M+/M–/5 Jahre |
| CX-8181L | AL | | 1 | 23 | X | | 1 | 8 | | X | | | | LCD | 8 + 1 | | X | | | |

| Hersteller Typ | Rechenlogik | Anzahl der Klammern | Anzahl der Funktionen | Anzahl der Tasten | Tastenbeleg. einfach | Tastenbeleg. mehrfach | Zahl der Speicher | Zahl der Stellen | Komma Fest- | Komma Fließ- | Komma Gleit- | techn.-wiss. | kaufmännisch | Anzeige Art | Anzeige Stellen | Stromversorgung Akku | Stromversorgung Batterie | Stromversorgung Netz | Ergänzungen | Besonderheiten |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| noch | | | | | | | | | | | | | | | | | | | | |
| ● **Sanyo** | | | | | | | | | | | | | | | | | | | | |
| CX-8191 | AL | | 1 | 22 | X | | 1 | 8 | | X | | | | LED | 8 + 1 | X | X | X | | %/+/_/M+ |
| CX-8192 | AL | | 1 | 22 | X | | 1 | 8 | | X | | | | DIG | 8 + 1 | X | X | X | | %/+/_/M+ |
| CY-1100P | ARL | | | 27 | X | | 1 | 10 | X | X | | | X | DIG | 10 + 1 | | | X | Drucker | Tischrechner/%/+/_/M+/M– |
| CY-2200P | ARL | | | 27 | X | | 1 | 12 | X | X | | | X | DIG | 12 + 1 | | | X | wie 1100P | wie 1100P |
| CY-2150 | ARL | | 1 | 23 | X | | 1 | 12 | X | X | | | X | DIG | 12 + 1 | | | X | | Tischrechner/%/M+ |
| CY-2166DP | ARL | | | 27 | X | | 1 | 12 | X | X | | | X | DIG | 12 + 1 | | | X | Drucker | Tischrechner/%/M+/M– |
| CZ-0111 | AL | 2 | 15 | 40 | X | | 1 | 8 | | | X | X | | LED | 10 + 2 | | X | | | π/+/_/M+/rad |
| CZ-0123 | AL | 2 | 14 | 37 | | X | 1 | 8 | | X | X | X | | DIG | 10 + 2 | X | X | X | | π/+/_/M+/rad/Polar |
| CZ-2171 | AL | 2 | 13 | 37 | X | | 1 | 10 | | X | X | X | | LED | 12 + 2 | X | | X | | π/+/_/rad |
| CZ-2172 | AL | | 17 | 35 | X | | 1 | 10 | | X | X | X | | LED | 12 + 2 | | X | X | | π/+/_/M+/rad |
| CZ-8100 | AL | | 12 | 26 | | X | 1 | 8 | | X | | X | | LED | 8 + 1 | X | | X | | π/+/_/M+/rad |
| CZ-8102 | AL | | 12 | 26 | | X | 1 | 8 | | X | | X | | LED | 8 + 1 | | X | | | π/+/_/M+/rad |
| CZ-8107F | AL | | 14 | 20 | | X | 1 | 8 | X | X | | | X | LED | 8 + 1 | | X | | | +/_/%/Δ%/M+/M + $x^2$/Statistik/Finanztechnik |
| CZ-8114L | AL | 2 | 45 | 38 | | X | 1 | 10 | | X | X | X | | LCD | 10 + 2 | | X | | | π/+/_/M+/M–/Statistik |
| CZ-8124 | AL | | 10 | 22 | | X | 1 | 8 | | X | X | X | | DIG | 7 + 2 | | X | X | | π/M+/rad |
| ICC-121 | ARL | | | 18 | X | | | | X | X | | | X | LED | 12 + 1 | | | X | | Tischrechner |
| ICC-141 | ARL | | | 23 | X | | 1 | 14 | X | X | | | X | LED | 14 + 1 | | | X | | Tischrechner/M+ |
| ICC-162 | ARL | | 1 | 29 | X | | 2 | 16 | X | X | | | X | LED | 16 + 1 | | | X | | Tischrechner/%/M+/M– |
| ICC-163P | ARL | | 1 | 33 | X | | 2 | 16 | X | X | | | X | | | | | X | Drucker (16 Stellen) | Tischrechner/%/M+/M– |
| ICC-801 | ARL | | | 18 | X | | | | X | X | | | X | LED | 8 + 1 | | X | X | | Tischrechner |
| ICC-805NE | ARL | | | 18 | X | | | | | X | | | X | DIG | 8 + 1 | | X | X | | Tischrechner |
| ICC-1415P | ARL | | | 29 | X | | 1 | 14 | X | X | | | X | | | | | X | Drucker (14 Stellen) | Tischrechner/M+/M– |
| ● **SERD** | | | | | | | | | | | | | | | | | | | | |
| 21DP-N | AL | | | 29 | X | | 1 | 10 | X | X | | | X | DIG | 10 + 1 | X | X | X | Drucker | kleiner Tischrechner/%/+/_/M+/M– |
| 212PD | AL | | | 29 | X | | 1 | 12 | X | X | | | X | DIG | 12 + 1 | X | X | X | wie 21 | wie 21 |
| 60P | ARL | | | 27 | X | | 1 | 12 | X | X | | | X | | | | | X | Drucker | Tischrechner |
| 61DP | ARL | | | 27 | X | | 1 | 12 | X | X | | | X | DIG | 12 + 1 | | | X | wie 60P | wie 60P |
| 71A | ARL | | | 27 | X | | 1 | 12 | X | X | | | X | DIG | 12 + 1 | | | X | Drucker | Tischrechner/%/+/_/M+/M– |
| 125 | ARL | | | 27 | X | | 1 | 12 | X | X | | | X | DIG | 12 + 1 | X | X | X | Drucker | wie 61DP |
| Aachen-LC | AL | 2 | 45 | 38 | | X | 1 | 10 | | X | X | X | | LCD | 10 + 2 | | X | | | π/+/_/M+/M–/Statistik |
| Aachen | AL | 2 | 14 | 37 | | X | 1 | 8 | | X | X | X | | DIG | 10 + 2 | X | X | X | | π/+/_/M+/rad |
| Augsburg | AL | | 1 | 23 | X | | 1 | 8 | | X | | | | LCD | 8 + 1 | | X | | | %/M+/M– |
| Berlin | AL | | 1 | 21 | X | | | | | X | | | | DIG | 8 + 1 | X | X | X | | %/+/_ |
| Bonn | AL | | | 18 | X | | | | | X | | | | LED | 8 + 1 | | X | X | | % |
| Bremen | AL | 1 | 3 | 28 | X | | 1 | 8 | | X | | | | DIG | 8 + 1 | X | X | X | | %/+/_/M+ |
| Düsseldorf | AL | | | 20 | X | | 1 | 8 | X | X | | | | DIG | 8 + 1 | | X | X | | % |
| Düsseldorf S | AL | | | 23 | X | | 1 | 8 | | X | | | | DIG | 8 + 1 | X | X | X | | %/+/_/M+ |
| Frankfurt | AL | | 1 | 23 | X | | 1 | 8 | | X | | | | LCD | 8 + 1 | | X | | | %/M+/M– |
| Futura | AL | | | 20 | X | | | | | X | | | | LCD | 8 + 1 | | X | | | %/+/_/1000 Stunden |
| Hamburg | AL | | 3 | 23 | X | | 1 | 8 | | X | | | | DIG | 8 + 1 | | X | X | | %/+/_/M+ |
| Hannover | AL | | 10 | 35 | X | | 1 | 8 | | X | X | X | | DIG | 7 + 2 | X | X | X | | π/M+/M–/rad |
| Kassel | AL | | 1 | 23 | X | | 1 | 8 | | X | | | | LCD | 8 + 1 | | X | | | %/M+/M– |
| Mini | AL | | 1 | 23 | X | | 1 | 8 | | X | | | | LCD | 8 + 1 | | X | | | %/M+/M– |
| München | AL | | | 19 | X | | | | | X | | | | DIG | 8 + 1 | | X | X | | % |
| Riviera | AL | | | 23 | X | | 1 | 8 | | X | | | | LCD | 8 + 1 | | X | | | rote LCD-Anzeige |
| Stuttgart | AL | | 12 | 22 | | X | 1 | 8 | | X | X | X | | DIG | 8 + 1 | | X | X | | π/+/_/M+/rad |
| Ulm | AL | 1 | 3 | 28 | X | | 1 | 8 | | X | | | | DIG | 8 + 1 | X | X | X | | π/%/M+ |
| ● **SHARP** | | | | | | | | | | | | | | | | | | | | |
| CS-239 | ARL | | | 23 | X | | 1 | 12 | X | X | | | X | DIG | 12 + 1 | | | X | | Tischrechner/M+/M– |
| CS-243V | ARL | | | 23 | X | | 2 | 14 | X | X | | | X | DIG | 14 + 1 | | | X | | Tischrechner/M+/M– |
| CS-523 | ARL | | | 20 | X | | | | X | X | | | X | | | | | X | Drucker (12 Stellen) | Tischrechner/% |
| CS-626 | ARL | | | 24 | X | | | | X | X | | | X | | | | | X | wie 523 | wie 523 |
| CS-628 | ARL | | | 25 | X | | | | X | X | | | X | DIG | 12 + 1 | | | X | wie 523 | wie 523 |
| CS-1057 | ARL | | | 19 | X | | | | X | X | | | X | | | | | X | Drucker (10 Stellen) | Tischrechner/% |

| Hersteller Typ | Rechenlogik | Anzahl der Klammern | Anzahl der Funktionen | Anzahl der Tasten | Tastenbeleg. einfach | Tastenbeleg. mehrfach | Zahl der Speicher | Zahl der Stellen | Komma Fest- | Komma Fließ- | Komma Gleit- | techn.-wiss. | kaufmännisch | Anzeige Art | Anzeige Stellen | Stromversorgung Akku | Stromversorgung Batterie | Stromversorgung Netz | Ergänzungen | Besonderheiten |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| noch | | | | | | | | | | | | | | | | | | | | |
| • SHARP | | | | | | | | | | | | | | | | | | | | |
| CS-1064 | ARL | | | 22 | X | | | | X | X | | | X | DIG | 10 + 1 | | | X | wie 1057 | wie 1057 |
| CS-1102 | ARL | | | 19 | X | | 1 | 10 | X | X | | | X | DIG | 10 + 1 | | | | | Tischrechner/% |
| CS-1109 | ARL | | 1 | 27 | X | | 1 | 10 | X | X | | | X | DIG | 10 + 1 | | | X | | Tischrechner/%/M+/M– |
| CS-1151 | ARL | | | 21 | X | | 1 | 10 | X | X | | | X | | | | | X | Drucker (10 Stellen) | Tischrechner |
| CS-1153 | ARL | | | 22 | X | | 1 | 10 | X | X | | | X | | | | | X | wie 1151 | Tischrechner/% |
| CS-1154B | ARL | | 2 | 24 | X | | 1 | 10 | X | X | | | X | | | | | X | Drucker (10 Stellen) | Tischrechner/% |
| CS-1164B | ARL | | 2 | 24 | X | | 1 | 10 | X | X | | | X | DIG | 10 + 1 | | | X | wie 1154B | wie 1154B |
| CS-2101 | ARL | | | 21 | X | | 1 | 12 | | X | | | X | DIG | 12 + 1 | | | X | | Tischrechner/M+/+/– |
| CS-2106 | ARL | | 1 | 25 | X | | 1 | 12 | X | X | | | X | DIG | 12 + 1 | | | X | | Tischrechner/%/M+/M– |
| CS-2109 | ARL | | 1 | 29 | X | | 1 | 12 | X | X | | | X | DIG | 12 + 1 | | | X | | wie 2106/+/– |
| CS-2151 | ARL | | | 26 | X | | 1 | 12 | X | X | | | X | DIG | 12 + 1 | | | X | Drucker | Tischrechner/%/M+/M– |
| CS-2152 | ARL | | | 25 | X | | 1 | 12 | X | X | | | X | | | | | X | wie 2151 (12 Stellen) | wie 2151 |
| CS-2156A | ARL | | 2 | 25 | X | | 1 | 12 | X | X | | | X | | | | | X | wie 1154B (12 Stellen) | wie 1154B |
| CS-2166A | ARL | | 2 | 35 | X | | 1 | 12 | X | X | | | X | DIG | 12 + 1 | | | X | wie 2156A | wie 2156A |
| CS-2182 | ARL | | 2 | 28 | X | | 1 | 12 | X | X | | | X | DIG | 12 + 1 | | | X | Drucker | Tischrechner/% |
| CS-4101 | ARL | | | 29 | X | | 1 | 14 | X | X | | | X | DIG | 14 + 1 | | | X | | Tischrechner/%/M+/M– |
| CS-4151 | ARL | | 1 | 29 | X | | 1 | 14 | X | X | | | X | | | | | X | Drucker (14 Stellen) | Tischrechner/%/+/–/M+/M– |
| CS-4152 | ARL | | 1 | 29 | X | | 1 | 14 | X | X | | | X | DIG | 14 + 1 | | | X | wie 4151 | wie 4151 |
| CS-4251 | ARL | | | 34 | X | | 2 | 14 | X | X | | | X | | | | | | wie 4151 | wie 4151 |
| CS-4663 | ARL | | 1 | 34 | X | | 3 | 14 | X | X | | | X | DIG | 14 + 1 | | | X | wie 4151 | wie 4151 |
| CS-6301 | ARL | | 1 | 37 | X | | 3 | 16 | X | X | | | X | DIG | 16 + 1 | | | X | | Tischrechner/%/M+/M– |
| CS-6401 | ARL | | 1 | 36 | X | | 4 | 16 | X | X | | | X | DIG | 16 + 1 | | | X | | wie 6301 |
| CT-510 | AL | | | 31 | | X | 1 | 8 | | X | | | | LCD | 8 + 1 | | X | | | %/M+/M–/Stoppuhr/Kalender/Sensortasten |
| CT-550 | AL | | 1 | 24 | | X | 1 | 8 | | X | | | | LCD | 8 + 1 | | X | | | %/M+/M–/Uhr/Kalender/17 Monate |
| CT-650 | AL | | 1 | 25 | | X | 1 | 8 | | X | | | | LCD | 8 + 1 | | X | | | Tischrechner/wie CT-550/Sensortasten |
| CT-4266 | ARL | | | 32 | X | | 2 | 12 | X | X | | | X | DIG | 12 + 1 | | | X | Drucker | Tischrechner/% |
| EL-203 | AL | | 1 | 20 | X | | 1 | 8 | | X | | | | LCD | 8 + 1 | | X | | | %/150 Stunden |
| EL-207 | AL | | 1 | 20 | X | | 1 | 8 | | X | | | | LCD | 8 + 1 | | X | | | % |
| EL-805M | AL | | | 21 | X | | 1 | 8 | | X | | | | LCD | 8 + 1 | | X | | | +/–/M+ |
| EL-805S | AL | | | 18 | X | | | | | X | | | | LCD | 8 + 1 | | X | | | |
| EL-817S | ARL | | | 21 | X | | 1 | 8 | | X | | | X | DIG | 8 + 1 | | | X | | Tischrechner/+/–/M+ |
| EL-1057 | ARL | | | 19 | X | | | | X | X | | | X | | | | | X | Drucker (10 Stellen) | Tischrechner/% |
| EL-1064 | ARL | | | 22 | X | | 1 | 10 | | X | | | X | DIG | 10 + 1 | | | X | wie 1057 | Tischrechner/%/M+ |
| EL-1071 | ARL | | | 19 | X | | | | X | X | | | X | | | | | X | Drucker | Tischrechner/% |
| EL-1100 | AL | | 15 | 25 | | X | 1 | 10 | | X | X | X | | DIG | 12 + 2 | X | X | X | | +/–/M+/rad/Statistik |
| EL-1101 | ARL | | | 21 | X | | 1 | 10 | X | X | | | X | DIG | 10 + 1 | X | X | X | | %/M+/M– |
| EL-1105 | AL | | 3 | 30 | X | | 1 | 10 | X | X | | | X | DIG | 10 + 1 | X | X | X | | +/–/M+/M–/Finanzrechner |
| EL-1110 | AL | | 1 | 25 | X | | 1 | 10 | X | X | | | | LCD | 10 + 1 | | X | | | %/M+/M–/750 Stunden |
| EL-1114 | AL | | 1 | 26 | X | | 1 | 10 | X | X | | | | LCD | 10 + 1 | | X | | | %/+/–/M+/M–/500 Std. |
| EL-1153 | ARL | | | 21 | X | | 1 | 10 | X | X | | | X | | | | | X | Drucker (10 Stellen) | Tischrechner/% |
| EL-1163 | ARL | | | 19 | X | | | | X | X | | | X | DIG | 10 + 1 | X | | X | wie 1153 | % (Taschenrechner) |
| EL-1167 | ARL | | | 28 | X | | 2 | 10 | X | X | | | X | LCD | 10 + 1 | | X | | Drucker | kleiner Tischrechner/%/+/–/M+/M– |
| EL-2158A | ARL | | | 25 | X | | 1 | 12 | X | X | | | X | | | | | X | Drucker | Tischrechner/%/M+/M– |
| EL-2168A | ARL | | | 25 | X | | 1 | 12 | X | X | | | X | DIG | 12 + 1 | | | X | wie 2158A | wie 2158A |
| EL-5000 | AL | 2 | 27 | 40 | | X | 1 | 10 | | X | X | X | | DIG | 12 + 2 | X | X | X | | π/%/+/–/M+/Hyp. Statistik/Bruchrechnung |
| EL-5804 | AL | | 12 | 30 | | X | 1 | 8 | | X | X | X | | LCD | 8 + 2 | | X | | | π/M+/Querformat |
| EL-5805 | AL | | 20 | 35 | | X | 1 | 8 | | X | X | X | | LCD | 8 + 2 | | X | | | π/+/–/M+/rad/Polar/Statistik |
| EL-5806 | AL | | 24 | 37 | | X | 1 | 8 | | X | X | X | | LCD | 8 + 2 | | X | | | wie 5805 |
| EL-5808 | AL | 1 | 28 | 35 | | X | 1 | 8 | | X | X | X | | LCD | 10 + 2 | | X | | | π/+/–/M+/Δ %/rad/Hyp./Statistik |
| EL-5809 | AL | | 18 | 33 | | X | 1 | 8 | | X | X | X | | LCD | 7 + 2 | | X | | | π/+/–/M+/Querformat/600 Stunden |
| EL-5810 | AL | 1 | 18 | 28 | | X | 1 | 8 | | X | X | X | | LCD | 8 + 2 | | X | | | π/+/–/Δ %/Statistik/600 Stunden |

| Hersteller Typ | Rechenlogik | Anzahl der Klammern | Anzahl der Funktionen | Anzahl der Tasten | Tastenbeleg. einfach | Tastenbeleg. mehrfach | Zahl der Speicher | Zahl der Stellen | Komma Fest- | Komma Fließ- | Komma Gleit- | techn.-wiss. | kaufmännisch | Anzeige Art | Anzeige Stellen | Stromversorgung Akku | Stromversorgung Batterie | Stromversorgung Netz | Ergänzungen | Besonderheiten |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| noch | | | | | | | | | | | | | | | | | | | | |
| ● **SHARP** | | | | | | | | | | | | | | | | | | | | |
| EL-8000R | AL | | 1 | 20 | X | | | | | X | | | | DIG | 8 + 1 | X | X | X | | % |
| EL-8005S | AL | | | 19 | X | | | | | X | | | | DIG | 8 + 1 | X | X | X | | % |
| EL-8008 | AL | | | 19 | X | | | | | X | | | | LCD | 8 + 1 | | X | | | %/+/_/100 Stunden |
| EL-8009 | AL | | | 17 | X | | | | | X | | | | LCD | 8 + 1 | | X | | | |
| EL-8010 | AL | | | 19 | X | | | | | X | | | | LCD | 8 + 1 | | X | | | %/+/_ |
| EL-8019 | AL | | 1 | 20 | X | | | | | X | | | | LCD | 8 + 1 | | X | | | %/1 200 Stunden |
| EL-8024 | AL | | 1 | 20 | X | | 1 | 8 | | X | | | | LCD | 8 + 1 | | X | X | | M+ |
| EL-8026 | AL | | 1 | 20 | X | | | | | X | | | | LCD | 8 + 1 | X | | | | %/Solarzellen |
| EL-8028 | AL | | 1 | 20 | X | | | | | X | | | | LCD | 8 + 1 | | X | | | %/1 200 Stunden |
| EL-8029 | AL | | 1 | 20 | X | | | | | X | | | | LCD | 8 + 1 | | X | | | %/500 Stunden/Querformat |
| EL-8039 | AL | | | 20 | X | | 1 | 8 | | X | | | | LCD | 8 + 1 | | X | | | %/Sensortasten |
| EL-8100S | AL | | 1 | 22 | X | | 1 | 8 | | X | | | | DIG | 8 + 1 | | X | X | | %/+/_/M+ |
| EL-8102 | AL | | 1 | 22 | X | | 1 | 8 | | X | | | | DIG | 8 + 1 | X | X | X | | %/+/_/M+ |
| EL-8118 | AL | | 4 | 30 | X | | 1 | 8 | X | X | | | | DIG | 8 + 1 | X | X | X | | π/%/+/_/M+/M– |
| EL-8121 | AL | | 1 | 22 | X | | 1 | 8 | | X | | | | LCD | 8 + 1 | | X | | | %/M+/M–/1 200 Stunden |
| EL-8128 | AL | | 1 | 25 | X | | 1 | 8 | | X | | | | LCD | 8 + 1 | | X | | | %/M+/M–/1 200 Stunden |
| EL-8130 | AL | | 1 | 24 | X | | 1 | 8 | | X | | | | LCD | 8 + 1 | | X | | | %/M+/M–/750 Stunden/Sensortasten |
| EL-8133 | AL | | 1 | 24 | X | | 1 | 8 | | X | | | | LCD | 8 + 1 | | X | | | %/M+/M–/1500 Stunden |
| EL-8140 | AL | | | 24 | X | | 1 | 8 | | X | | | | LCD | 8 + 1 | | X | | | %/+/_/M+/M–/Permanentspeicher |
| EL-8144 | AL | | | 24 | | X | 1 | 8 | | X | | | | LCD | 8 + 1 | | X | | | %/M+/M–/Datum/Stoppuhr/Wecker/Sensor-tasten |
| EL-8145 | AL | | | 24 | X | | 1 | 8 | | X | | | | LCD | 8 + 1 | | X | | | wie EL-8140, Querformat |
| EL-8146 | AL | | 1 | 25 | X | | 1 | 8 | | X | | | | | | | | | | %/M+/M– |
| EL-8149 | AL | | 1 | 25 | X | | 1 | 8 | | X | | | | LCD | 8 + 1 | | X | | | %/M+/M–/1500 Stunden |
| EL-8150 | AL | | | 24 | X | | 2 | 8 | | X | | | | LCD | 8 + 1 | | X | | | %/M+/M–/Permanentspeicher/1000 Stunden |
| EL-8151 | AL | | | 19 | X | | 1 | 8 | | X | | | | DIG | 8 + 1 | X | | X | Drucker | % |
| EL-8152 | AL | | | 26 | X | | 2 | 8 | | X | | | | LCD | 8 + 1 | | X | | | wie EL-8150/1,6 mm flach |
| EL-8200 | AL | | | 20 | | X | | | X | X | | | X | DIG | 8 + 1 | | X | X | | %/Finanzrechner |
| EL-8245 | AL | | | 28 | | X | 2 | 8 | | X | | | | LCD | 8 + 1 | | X | | | %/M+/M–/Wecker/Stoppuhr/Kalender/6 Monate |
| PC-1802 | AL | | 10 | 25 | | X | 1 | 8 | | X | | X | | DIG | 8 + 1 | | X | X | | π/+/_/M+/M– |
| ● **SILVER-RRED** | | | | | | | | | | | | | | | | | | | | |
| SR83 | AL | | 10 | 19 | | X | 1 | 8 | | X | X | X | | DIG | 7 + 2 | X | | | | π/M+/M– |
| SR84 | AL | | 12 | 26 | | X | 1 | 8 | | X | | X | | DIG | 8 + 1 | X | | X | | π/+/_/M+/rad |
| LCD-TIME | AL | | 1 | 32 | X | | 1 | 8 | | X | | | | LCD | 8 + 1 | | X | | | %/+/_/M+/M–/Uhr/Wecker/2 500 Stunden |
| SR LC-2000 | AL | | 1 | 24 | X | | 1 | 8 | | X | | | | LCD | 8 + 1 | | X | | | %/+/_/M+/M–/2 500 Stunden |
| SR LCD N-ESR | AL | 2 | 22 | 35 | | X | 1 | 8 | | X | X | X | | LCD | 7 + 2 | | X | | | π/+/_/M+/M–/Statistik/Umrechnungen |
| ● **Sinclair** | | | | | | | | | | | | | | | | | | | | |
| Cambridge | AL | | | 18 | X | | | | | X | | | | LED | 8 + 1 | | X | | | |
| Cambridge Memory | AL | | | 18 | X | | 1 | 8 | | X | | | | LED | 8 + 1 | | X | | | |
| Executive Memory | ARL | | | 19 | X | | 1 | 8 | | X | | | | LED | 8 + 1 | | X | | | |
| Oxford 100 | AL | | | 19 | X | | | | | X | | | | LED | 8 + 1 | | X | X | | |
| Oxford 200 | AL | | | 19 | X | | 1 | 8 | | X | | | | LED | 8 + 1 | | X | X | | %/M+/M– |
| Oxford 300 | AL | | 10 | 19 | | X | 1 | 8 | | X | X | X | | LED | 7 + 2 | | X | X | | π/M+/M– |
| Scientific | ARL | | 8 | 18 | | X | | | | X | X | X | | LED | 7 + 2 | | X | | | |
| ● **SUNTIE** | | | | | | | | | | | | | | | | | | | | |
| CD-300 | AL | | | 22 | X | | 1 | 8 | | X | | | | LCD | 8 + 1 | | X | | | %/M+/M– |
| ● **Superlectron** | | | | | | | | | | | | | | | | | | | | |
| 8 | AL | | | 18 | X | | | | | X | | | | LED | 8 + 1 | | X | X | | |
| 8A | AL | | 3 | 24 | X | | 1 | 8 | | X | | | | LED | 8 + 1 | | X | X | | %/+/_ |
| 8M | AL | | | 24 | X | | 1 | 8 | | X | | | | LED | 8 + 1 | | X | X | | %/M+/M– |
| 8P | AL | | | 18 | X | | | | | X | | | | LED | 8 + 1 | | X | X | | % |
| 8S | AL | | 11 | 26 | | X | 1 | 8 | | X | | X | | LED | 8 + 1 | | X | X | | π/+/_/M+/rad |
| 12M | AL | | | 26 | X | | 1 | 12 | | X | | | | LED | 12 + 1 | | X | X | | %/+/_/M+/M– |
| 12P | ARL | | | 25 | X | | 1 | 12 | X | X | | | X | | | | | X | Drucker (12 Stellen) | Tischrechner/%/M+/M– |

| Hersteller Typ | Rechenlogik | Anzahl der Klammern | Anzahl der Funktionen | Anzahl der Tasten | Tastenbeleg. einfach | Tastenbeleg. mehrfach | Zahl der Speicher | Zahl der Stellen | Komma Fest- | Komma Fließ- | Komma Gleit- | techn.-wiss. | kaufmännisch | Anzeige Art | Anzeige Stellen | Stromversorgung Akku | Stromversorgung Batterie | Stromversorgung Netz | Ergänzungen | Besonderheiten |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| noch | | | | | | | | | | | | | | | | | | | | |
| • **Superlectron** | | | | | | | | | | | | | | | | | | | | |
| 12PD | ARL | | | 27 | X | | 1 | 12 | X | X | | | X | DIG | 12 + 1 | | | X | wie 12P | wie 12P |
| 608 | AL | | | 20 | X | | | | | X | | | | DIG | 8 + 1 | | X | X | | +/– |
| 1200 | ARL | | | 25 | X | | 1 | 12 | X | X | | | | DIG | 12 + 1 | | | X | | Tischrechner/%/M+/M– |
| 8010D | ARL | | | 19 | X | | | | | X | | | | DIG | 8 + 1 | | | X | | Tischrechner/% |
| VL12 | AL | | 1 | 22 | X | | | | | X | | | | DIG | 8 + 1 | | X | X | | % |
| VL22 | AL | | 3 | 24 | X | | 1 | 8 | | X | | | | DIG | 8 + 1 | | X | X | | %/+/– |
| VL25 | AL | | 8 | 32 | X | | 1 | 8 | | X | X | X | | DIG | 10 + 2 | | X | X | | +/–/M+ |
| VL30 | AL | | 10 | 26 | | X | 1 | 8 | | X | | X | | DIG | 8 + 1 | | X | X | | π/+/–/M+/rad |
| VL35 | AL | | 10 | 19 | | X | 1 | 8 | | X | X | X | | DIG | 7 + 2 | | X | X | | π/+/–/M+/M–/rad |
| VL40 | AL | | 18 | 35 | X | | 1 | 10 | | X | X | X | | DIG | 12 + 2 | | X | X | | π/+/–/M+/rad/Hyp. |
| VL42 | AL | 1 | 14 | 25 | | X | 1 | 10 | | X | X | X | | LED | 12 + 2 | | X | X | | π/+/–/M+/M–/M x/M ÷/rad |
| VL810 | ARL | | | 17 | X | | | | | X | | | | DIG | 8 + 1 | | X | X | | |
| VL816 | AL | | | 20 | X | | | | | X | | | | DIG | 8 + 1 | | X | X | | % |
| VL852 | AL | | | 20 | X | | 1 | 8 | | X | | | | DIG | 8 + 1 | | X | X | | % |
| VLF-65 | AL | | 15 | 20 | | X | 3 | 8 | | X | | | | DIG | 8 + 1 | | X | X | | %/Δ%/+/–/Finanzrechner |
| • **Systema** | | | | | | | | | | | | | | | | | | | | |
| LC2600 | AL | | 4 | 28 | X | | 1 | 8 | | X | | | | LCD | 8 + 1 | | X | | | π/%/+/–/M+/M–/2000 Stunden |
| LC3500 | AL | 2 | 28 | 40 | | X | 1 | 10 | | X | X | X | | LCD | 10 + 2 | | X | | | π/+/–/M+/rad/Hyp./Statistik/Δ%/1000 Stunden |
| LC4500 | AL | | 1 | 30 | X | | 1 | 8 | | X | | | | LCD | 8 + 1 | | X | | | %/M+/M–/Datum/Stoppuhr/Wecker |
| • **TEALTRONIC** | | | | | | | | | | | | | | | | | | | | |
| PMP1200S | ARL | | | 24 | X | | 1 | 12 | X | X | | | X | | | | | X | Drucker | Tischrechner/M+/M– |
| T10/2SRD | ARL | | 1 | 25 | X | | 1 | 12 | X | X | X | | X | DIG | 12 + 2 | | | X | | Tischrechner/%/M+/M– |
| T121D | AL | | 14 | 35 | X | | 1 | 12 | | X | X | X | | DIG | 12 + 2 | | | X | | Tischrechner/π/M+/rad |
| T122P | ARL | | 2 | 29 | X | | | | X | X | | | X | | | | | X | Drucker | Tischrechner/%/+/– |
| T123PD | ARL | | | 26 | X | | 1 | 12 | X | X | | | X | DIG | 12 + 1 | | | X | Drucker | Tischrechner/M+/M– |
| T4517 | AL | | 1 | 20 | X | | | | | X | | | | DIG | 8 + 1 | | X | | | %/Querformat |
| T4519 | AL | | 1 | 26 | X | | 1 | 8 | | X | | | | DIG | 8 + 1 | | X | | | π/%/M+/M–/Querformat |
| Tic63 | AL | | 1 | 26 | X | | 1 | 8 | X | X | | | | DIG | 8 + 1 | | X | | | π/%/M+/M– |
| Tic85 | AL | | 11 | 26 | | X | 1 | 8 | | X | | X | | DIG | 8 + 1 | | X | | | π/+/–/M+/rad |
| Tic86 | AL | 2 | 14 | 25 | | X | 1 | 8 | | X | X | X | | DIG | 10 + 2 | | X | | | π/+/–/M+/M–/M x/M ÷/rad |
| TicFX | AL | | 15 | 20 | | X | | | | X | | | X | DIG | 8 + 1 | X | X | X | | %/Δ%/+/–/Finanzrechner |
| • **Texas Instruments** | | | | | | | | | | | | | | | | | | | | |
| SR-10 | AL | | 3 | 23 | X | | | | | | X | X | | LED | 10 + 2 | X | | | | +/– |
| SR-11 | AL | | 3 | 24 | X | | | | | | X | X | | LED | 10 + 2 | X | | | | π/+/– |
| SR-16 | AL | | 8 | 31 | X | | 1 | 8 | | | X | X | | LED | 10 + 2 | X | | | | +/– |
| SR-22 | AL | | | 34 | X | | 1 | | | | X | X | | LED | 12 + 2 | X | | X | | Tischrechner/Arithmetik zur Basis 8, 10 und 16 |
| SR-40 | AOS | 15 | 14 | 40 | X | | 1 | 11 | | | X | X | | LED | 8 + 1 | X | | | | π/%/rad |
| SR-50 | ALH | | 17 | 40 | X | | 1 | 13 | | | X | X | | LED | 12 + 2 | X | | | | π/Hyp./M+/rad |
| SR-51 | ALH | | 35 | 40 | | X | 3 | 13 | X | | X | X | | LED | 12 + 2 | X | | | | π/%/Δ%/+/–/rad/M+/M x/Statistik/Umwandlungen |
| SR-51-II | AOS | 9 | 35 | 40 | | X | 3 | 12 | X | | X | X | | LED | 10 + 2 | X | | | | wie SR-51 |
| TI-15 | AOS | | 3 | | | | 1 | | | | | | | | | | X | | | Schulrechner/π/% |
| TI-25 | ALH | 3 | 18 | 40 | X | | 1 | 9 | | | X | X | | LCD | 8 + 1 | | X | | | π/%/1000 Betriebsstunden |
| TI-30 | AOS | 15 | 14 | 40 | X | | 1 | 11 | | | X | X | | LED | 8 + 1 | X | X | X | | π/%/rad |
| TI-33 | AL | 1 | 14 | 40 | X | | 3 | 11 | | | X | X | | LED | 8 + 1 | X | X | X | | π/rad |
| TI-41 | AOS | 15 | 19 | 40 | | X | 1 | 11 | | X | | | X | LED | 8 + 1 | X | | | | %/Δ%/Statistik/Finanztechnik |
| TI-44 | AL | | 10 | 40 | | X | 1 | 8 | | X | | | X | LCD | 8 + 1 | | X | | | +/–/%/Δ%/M+/M–/Finanztechnik/Statistik/Permanentspeicher |
| TI-45 | AOS | 15 | 15 | 40 | X | | 1 | 11 | | | X | X | | DIG | 8 + 1 | X | | | | π/%/rad |
| TI-50 | AOS | 15 | 16 | 40 | | X | 2 | 8 | | X | X | X | | LCD | 7 + 2 | | X | | | π/+/–/%/M+/M–/1000 Stunden/Permanentspeicher |
| TI-450 | ARL | | | 26 | X | | 1 | 12 | X | X | | | X | LED | 12 | | | X | | Tischrechner/% |
| TI-1025 | AL | | | 24 | X | | 1 | 8 | | X | | | | DIG | 8 + 1 | | X | X | | %/+/–/M+/M– |
| TI-1030 | AL | | 1 | 24 | X | | 1 | 8 | | X | | | | LCD | 8 + 1 | | X | | | %/M+/M–/2000 Stunden/Etui |
| TI-1050 | AL | | 1 | 24 | X | | 1 | 8 | | X | | | | DIG | 8 + 1 | | X | X | | %/M+/M–/M x/M ÷/+/– |

| Hersteller Typ | Rechenlogik | Anzahl der Klammern | Anzahl der Funktionen | Anzahl der Tasten | Tastenbeleg. einfach | Tastenbeleg. mehrfach | Zahl der Speicher | Zahl der Stellen | Komma Fest- | Komma Fließ- | Komma Gleit- | techn.-wiss. | kaufmännisch | Anzeige Art | Anzeige Stellen | Stromversorgung Akku | Stromversorgung Batterie | Stromversorgung Netz | Ergänzungen | Besonderheiten |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| noch | | | | | | | | | | | | | | | | | | | | |
| • **Texas Instruments** | | | | | | | | | | | | | | | | | | | | |
| TI-1070 | AL | | 3 | 24 | X | | 1 | 8 | | X | | | | LCD | 8 + 1 | | X | | | $\pi$/+/_/3000 Stunden |
| TI-1200 | AL | | | 20 | X | | | | | X | | | | LED | 8 + 1 | | X | | | % |
| TI-1250 | AL | | | 24 | X | | 1 | 8 | | X | | | | LED | 8 + 1 | | X | | | %/M+/M–/+/_ |
| TI-1260 | AL | | | 24 | X | | | | X | | | | | LED | 8 + 1 | | X | | | %/Preisvergleiche |
| TI-1265 | AL | | | 24 | X | | 1 | 8 | | X | | | | DIG | 8 + 1 | | X | | | %/M+/M–/+/_ |
| TI-1270 | AL | | 3 | 24 | X | | 1 | 8 | | X | | | | LED | 8 + 1 | | X | | | $\pi$/+/_ |
| TI-1500 | AL | | | 19 | X | | | | | X | | | | LED | 8 + 1 | X | | | | % |
| TI-1600 | AL | | | 20 | X | | | | | X | | | | DIG | 8 + 1 | X | | | | % |
| TI-1650 | AL | | | 24 | X | | 1 | 8 | | X | | | | DIG | 8 + 1 | X | | X | | %/+/_/M+/M– |
| TI-1680 | AL | | | 24 | X | | 1 | 8 | | X | | | | DIG | 8 + 1 | X | | X | | %/+/_/Rückruf |
| TI-1700 | AL | | | 22 | X | | 1 | 8 | | X | | | | LCD | 8 + 1 | | X | | | %/+/_/M+/M–/Querformat/1000 Stunden |
| TI-1750 | AL | | 1 | 24 | X | | 1 | 8 | | X | | | | LCD | 8 + 1 | | X | | | %/M+/M–/Etui/2 000 Stunden |
| TI-1790 | AL | | 1 | 31 | X | | | | | X | | | | LCD | 8 + 1 | | X | | | %/Stoppuhr/Wecker/10 000 Stunden |
| TI-2500 | AL | | | 18 | X | | | | | X | | | | LED | 8 + 1 | X | | | | |
| TI-2500-II | AL | | | 19 | X | | | | | X | | | | LED | 8 + 1 | X | | | | % |
| TI-2550 | AL | | | 23 | X | | 1 | 8 | X | X | | | | LED | 8 + 1 | X | | X | | %/M+/M– |
| TI-2550-II | AL | | 3 | 27 | X | | 1 | 8 | X | X | | | | DIG | 8 + 1 | X | | X | | %/M+/M– |
| TI-2550-III | AL | | 3 | 28 | X | | 1 | 8 | | X | | | | DIG | 8 + 1 | X | | X | | %/M+/M–/+/_ |
| TI-2550-IV | AL | | | 24 | X | | 1 | 8 | | X | | | | DIG | 8 + 1 | X | | X | | %/+/_/Ablaufspeicher |
| TI-3500 | ARL | | | 17 | X | | | | X | X | | | | LED | 10 + 1 | | | X | | kleiner Tischrechner |
| TI-4000 | ARL | | | 22 | X | | 1 | 12 | X | X | | | | LED | 12 + 1 | | | X | | %/M+/M–/kleiner Tischrechner |
| TI-5015 | ARL | | | 22 | X | | | | X | X | | | X | | | | | X | Drucker (10 stellig) | Tischrechner/% |
| TI-5025 | AL | | | 24 | X | | 1 | 8 | | X | | | | DIG | 8 + 1 | X | | X | Drucker | %/M+/M– |
| TI-5040 | ARL | | | 28 | X | | 1 | 10 | X | X | | | X | DIG | 10 + 1 | | | X | Drucker | Tischrechner/%/M+/M– |
| TI-5050 | ARL | | | 21 | X | | | | X | X | | | X | | | X | | X | Drucker (9 stellig) | % |
| TI-5050M | ARL | | | 23 | X | | 1 | 10 | X | X | | | X | | | X | | X | Drucker (10 stellig) | % |
| TI-5100 | ARL | | | 22 | X | | 1 | 10 | X | X | | | X | DIG | 10 + 1 | | | X | | kleiner Tischrechner/%/M+/M– |
| TI-5200 | ARL | | | 22 | X | | 1 | 12 | X | X | | | X | DIG | 12 + 1 | | | X | | kleiner Tischrechner/%/M+/M– |
| TI-5220 | ARL | | | 31 | X | | 1 | 12 | X | X | | | X | DIG | 12 + 1 | | | X | Drucker | kleiner Tischrechner/%/M+/M– |
| TI-PROGRAMMER | AL | 15 | | 40 | X | | 1 | 11 | | X | | | | LED | 8 + 1 | X | | X | | Oktal, Hexadezimal, Umrechnungen, logische Operationen |
| • **TOSHIBA** | | | | | | | | | | | | | | | | | | | | |
| BC-1010BJ | AL | 1 | 1 | 20 | | X | | | | X | | | | DIG | 8 + 1 | | X | | | Black Jack-Spiel |
| BC-1014PV | AL | | | 30 | X | | 1 | 10 | X | X | | | | DIG | 10 + 1 | X | | X | Drucker | %/+/_/M+/M– |
| BC-1015 | ARL | | | 25 | X | | 1 | 10 | X | X | | | X | DIG | 10 + 1 | X | X | X | | kleiner Tischrechner/%/M+/M– |
| BC-1200P | ARL | | | 23 | X | | | | X | X | | | X | | | | | X | Drucker (12 Stellen) | Tischrechner/%/+/_ |
| BC-1210P | ARL | | | 23 | X | | 1 | 12 | X | X | | | X | | | | | X | wie 1200P | Tischrechner/%/M+ |
| BC-1215PV | ARL | | | 27 | X | | | | X | X | | | X | DIG | 12 + 1 | X | | X | Drucker | Tischrechner/%/+/_ |
| BC-1263P | ARL | | 1 | 30 | X | | 4 | 12 | X | X | | | X | | | | | X | wie 1200P | Tischrechner/%/M+ |
| BC-1264PV | ARL | | 1 | 30 | X | | 4 | 12 | X | X | | | X | DIG | 12 + 1 | | | X | wie 1200P | wie 1263P |
| BC-1270 | ARL | | 1 | 24 | X | | 4 | 12 | X | X | | | X | DIG | 12 + 1 | | | X | | Tischrechner/%/M+ |
| BC-1271P | ARL | | | 30 | X | | 4 | 12 | X | X | | | X | | | | | X | wie 1200P | Tischrechner/%/+/_/M+ |
| BC-1272PV | ARL | | | 30 | X | | 4 | 12 | X | X | | | X | DIG | 12 + 1 | | | X | wie 1200P | wie 1271P |
| BC-1281P | ARL | | | 27 | X | | | | X | X | | | X | | | | | X | Drucker (12 Stellen) | Tischrechner/+/_/% |
| BC-1282PV | ARL | | | 31 | X | | 1 | 12 | X | X | | | X | DIG | 12 + 1 | | | X | wie 1281 | wie 1281/M+/M– |
| BC-1283PV | ARL | | | 38 | X | | 1 | 12 | X | X | | | X | DIG | 12 + 1 | | | X | wie 1281 | wie 1282 |
| BC-1284PV | ARL | | 1 | 42 | X | | 1 | 12 | X | X | | | X | DIG | 12 + 1 | | | X | wie 1281 | wie 1283/$\Delta$% |
| BC-1285PC | ARL | | 1 | 42 | X | | 1 | 12 | X | X | | | X | DIG<br>LCD | 12 + 1<br>6 | | | X | wie 1281 | wie 1284/Uhr/Wecker/Kalender mit 6 stelliger LCD-Anzeige |
| BC-8018 | AL | | 1 | 20 | X | | | | | X | | | | DIG | 8 + 1 | | X | X | | $\pi$/% |
| BC-8111 | AL | | 1 | 24 | X | | 1 | 8 | | X | | | | DIG | 8 + 1 | | X | X | | $\pi$/%/M+/M– |

| Hersteller Typ | Rechenlogik | Anzahl der Klammern | Anzahl der Funktionen | Anzahl der Tasten | Tastenbeleg. einfach | Tastenbeleg. mehrfach | Zahl der Speicher | Zahl der Stellen | Komma Fest- | Komma Fließ- | Komma Gleit- | techn.-wiss. | kaufmännisch | Anzeige Art | Anzeige Stellen | Stromversorgung Akku | Stromversorgung Batterie | Stromversorgung Netz | Ergänzungen | Besonderheiten |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| noch | | | | | | | | | | | | | | | | | | | | |
| • **TOSHIBA** | | | | | | | | | | | | | | | | | | | | |
| BC-8112SR | AL | | 3 | 28 | X | | 1 | 8 | | X | | | | DIG | 8 + 1 | | X | X | | π/%/M+/M– |
| BC-8115S | AL | | 1 | 25 | | X | 1 | 8 | | X | | | | DIG | 8 + 1 | X | | X | | %/+/–/M+/M– |
| KT-1350 | AL | | 1 | 24 | X | | 1 | 8 | | X | | | | LCD | 8 + 1 | | X | | | %/M+/M–/Diktiergerät/3000 Stunden |
| LC-110PV | AL | | | 30 | X | | 1 | 10 | X | X | | | X | LCD | 10 + 1 | X | X | X | Drucker | kleiner Tischrechner/%/+/–/M+/M– |
| LC-821 | AL | | 1 | 24 | X | | 1 | 8 | | X | | | | LCD | 8 + 1 | | X | | | %/M+/M–/3500 Stunden |
| LC-826 | AL | | 1 | 24 | X | | 1 | 8 | | X | | | | LCD | 8 + 1 | | X | | | %/+/–/M+/M– |
| LC-829 | AL | | 1 | 24 | X | | 1 | 8 | | X | | | | LCD | 8 + 1 | | X | | | wie 826 |
| LC-830 | AL | | 1 | 24 | X | | 1 | 8 | | X | | | | LCD | 8 + 1 | | X | | | %/+/–/M+/M–/2000 Stunden |
| LC-831W | AL | | 1 | 26 | X | | 1 | 8 | | X | | | | LCD | 8 + 1 | | X | | | %/+/–/M+/M–/Uhr mit getrennter Anzeige/ 1000 Stunden |
| LC-832WA | AL | | 1 | 24 | X | | 1 | 8 | | X | | | | LCD | 8 + 1 | | X | | | %/+/–/M+/M–/Stoppuhr/Wecker mit getrennter Anzeige/1000 Stunden |
| LC-833WA | AL | | 1 | 25 | X | | 1 | 8 | | X | | | | LCD | 8 + 1 | | X | | | %/+/–/M+/M–/Kalender/Stoppuhr/Wecker/ 10 000 Stunden |
| LC-834SL | AL | | 1 | 25 | X | | 1 | 8 | | X | | | | LCD | 8 + 1 | | | | | %/+/–/M+/M–/Solarzellen |
| LC-834WA | AL | | 1 | 31 | X | | | | | X | | | | LCD | 8 + 1 | | X | | | %/+/–/M+/M–/Stoppuhr/Kalender/10 000 Std. |
| LC-835SR | AL | | 3 | 24 | X | | 1 | 8 | | X | | | | LCD | 8 + 1 | | X | | | %/M+/M–/2000 Stunden |
| LC-836MN | AL | | 1 | 30 | | X | 31 | 8 | | X | | | | LCD | 8 + 1 | | X | | | %/+/–/M+/M–/6-stellige Buchstabeneingabe/ 9000 Stunden |
| LC-838 | AL | | 1 | 24 | | X | 1 | 8 | | X | | | | LCD | 8 + 1 | | X | | | %/+/–/M+/Umrechnungen |
| LC-850M | AL | | 1 | 24 | X | | 1 | 8 | | X | | | | LCD | 8 + 1 | | X | | | %/+/–/M+/M–/2 000 Stunden |
| LC-851 | AL | | 1 | 24 | X | | 1 | 8 | | X | | | | LCD | 8 + 1 | | X | | | %/+/–/M+/M– |
| LC-852 | AL | | 1 | 24 | X | | 1 | 8 | | X | | | | LCD | 8 + 1 | | X | | | wie 850M |
| LC-853WA | AL | | 1 | 26 | | X | 1 | 8 | | X | | | | LCD | 8 + 1 | | X | | | wie 850M/Stoppuhr/Wecker/Kalender |
| LC-854WA | AL | | 1 | 25 | | X | 1 | 8 | | X | | | | LCD | 8 + 1 | | X | | | ähnlich 853WA |
| LC-1038MN | AL | | 1 | 35 | | X | 30 | 10 | | X | | | | LCD | 10 + 1 | | X | | | %/+/–/M+/M–/Buchstabeneingabe |
| LC-1280 | AL | | | 28 | X | | 1 | 12 | X | X | | | | LCD | 12 + 1 | | X | | | Tischrechner/%/Δ %/+/–/2500 Stunden |
| LC-1290C | AL | | | 28 | X | | 1 | 12 | X | X | | | | LCD | 12 + 6 | | X | | | Tischrechner/%/Δ %/+/–/Uhr mit extra Anzeige/ 2500 Stunden |
| LC-8000 | AL | | 1 | 25 | X | | 1 | 8 | | X | | | | LCD | 8 + 1 | | X | | | %/+/–/M+/M–/1000 Stunden |
| SC-7400 | AL | 1 | 14 | 43 | X | | 1 | 8 | | X | X | X | | DIG | 10 + 2 | | X | X | | π/+/–/M+/rad/Statistik |
| SC-7500 | AL | 2 | 27 | 43 | | X | 5 | 8 | | X | X | X | | DIG | 10 + 2 | | X | X | | π/+/–/M+/rad/Statistik |
| SLC-8100 | AL | | 14 | 30 | | X | 1 | 8 | | X | X | X | | LCD | 7 + 2 | | X | | | π/+/–/M+/M–/850 Stunden |
| SLC-8200 | AL | | 21 | 35 | | X | 1 | 8 | | X | X | X | | LCD | 7 + 2 | | X | | | π/+/–/M+/rad/1200 Stunden |
| SLC-8300 | AL | 2 | 24 | 40 | | X | 1 | 8 | | X | X | X | | LCD | 10 + 2 | | X | | | π/+/–/M+/%/Δ %/Statistik/2000 Stunden |
| • **TOSHO** | | | | | | | | | | | | | | | | | | | | |
| 720PD | ARL | | | 29 | X | | 2 | 12 | X | X | | | X | DIG | 12 + 1 | X | X | X | Drucker | kleiner Tischrechner/%/+/–/M+/M– |
| 3300PD | ARL | | 1 | 34 | X | | 1 | 12 | X | X | | | X | DIG | 12 + 1 | | | X | Drucker | Tischrechner/%/+/–/M+/M– |
| 3500D | ARL | | 1 | 34 | X | | 3 | 12 | X | X | | | X | DIG | 12 + 1 | | | X | Drucker | wie 3300PD |
| • **TOWA** | | | | | | | | | | | | | | | | | | | | |
| 120 | ARL | | | 29 | X | | 1 | 12 | X | X | | | X | DIG | 12 + 1 | X | | X | Drucker | kleiner Tischrechner/%/+/–/M+/M– |
| CS-504 | ARL | | | 33 | X | | 1 | 12 | X | X | | | X | DIG | 12 + 1 | | | X | Drucker | Tischrechner/%/M+/M– |
| • **TRICOM** | | | | | | | | | | | | | | | | | | | | |
| TC-1010LPD | AL | | | 29 | X | | 1 | 10 | X | X | | | X | LCD | 10 + 1 | X | X | X | Drucker | kleiner Tischrechner/%/M+/M– |
| TLC-8MCM | AL | | | 22 | | X | 1 | 8 | | X | | | | LCD | 8 + 1 | | X | | | %/M+/M–/Uhr |
| TLC-8MCR | AL | | 1 | 22 | | X | | | | X | | | | LCD | 8 + 1 | | X | | | %/+/–/Umrechnungen |
| TLC-8WTR | AL | | 1 | 31 | | X | 1 | 8 | | X | | | | LCD | 8 + 1 | | X | | | %/+/–/M+/M–/Stoppuhr/Wecker/Weltzeit/ Kalender |
| TLC-2100MR | AL | | 1 | 31 | | X | 21 | 8 | | X | | | | LCD | 8 + 1 | | X | | | %/+/–/M+/M–/alphanumerisch/Permanent |
| • **TRIUMPH ADLER** | | | | | | | | | | | | | | | | | | | | |
| 8K | ARL | | | 19 | X | | | | X | X | | | | DIG | 8 + 1 | | | X | | Tischrechner/% |

| Hersteller Typ | Rechenlogik | Anzahl der Klammern | Anzahl der Funktionen | Anzahl der Tasten | Tastenbeleg. einfach | Tastenbeleg. mehrfach | Zahl der Speicher | Zahl der Stellen | Komma Fest- | Komma Fließ- | Komma Gleit- | techn.-wiss. | kaufmännisch | Anzeige Art | Anzeige Stellen | Stromversorgung Akku | Stromversorgung Batterie | Stromversorgung Netz | Ergänzungen | Besonderheiten |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| noch | | | | | | | | | | | | | | | | | | | | |
| • **TRIUMPH ADLER** | | | | | | | | | | | | | | | | | | | | |
| 10XP | ARL | | | 24 | X | | | | X | X | | | X | | | | | X | Drucker (10 Stellen) | kleiner Tischrechner/%/+/– |
| 10XPD | ARL | | | 24 | X | | | | X | X | | | X | DIG | 10 + 1 | | | X | wie 10XP | wie 10XP |
| 12M | ARL | | | 22 | X | | 1 | 12 | X | X | | | | DIG | 12 + 1 | | | X | | Tischrechner/%/M+/M– |
| 12MK-AD | ARL | | 1 | 26 | X | | 1 | 12 | X | X | | | | DIG | 12 + 1 | X | | X | | Tischrechner/%/M+/M– |
| 12P | ARL | | | 24 | X | | | | X | X | | | | | | | | X | Drucker | Tischrechner/% |
| 12XPD | ARL | | | 24 | X | | | | X | X | | | X | DIG | 12 + 1 | | | X | Drucker | wie 10XPD |
| 80C | AL | | | 19 | X | | | | | X | | | | DIG | 8 + 1 | | X | | | % |
| 80CS | AL | | 3 | 22 | X | | | | X | X | | | | DIG | 8 + 1 | X | X | | | π/% |
| 80S | AL | | 1 | 22 | X | | | | | X | | | | DIG | 8 + 1 | | X | | | π/%/+/– |
| 81C | AL | | | 24 | X | | 1 | 8 | X | X | | | | DIG | 8 + 1 | | X | | | %/+/–/M+/M– |
| 81CS | AL | | 3 | 27 | X | | 1 | 8 | X | X | | | | DIG | 8 + 1 | X | X | | | π/%/Δ %/M+/M– |
| 81D | AL | | 1 | 25 | X | | 1 | 8 | | X | | | | DIG | 8 + 1 | | X | | | %/+/–/M+/M– |
| 81S | AL | | 1 | 27 | X | | 1 | 8 | X | X | | | | DIG | 8 + 1 | | X | | | π/%/+/–/M+/M– |
| 81SN | AL | | 1 | 27 | X | | 1 | 8 | X | X | | | | DIG | 8 + 1 | X | X | X | | wie 81S |
| 82M | AL | | | 20 | | X | 2 | 8 | | X | | | | DIG | 8 + 1 | | X | | | Bruchrechnung/Umrechnungen |
| 84F | AL | | | 20 | | X | 4 | 8 | X | X | | | X | DIG | 8 + 1 | | X | | | Statistik/Finanzen |
| 88T | AL | | 12 | 25 | | X | 1 | 8 | | X | | X | | DIG | 8 + 1 | | X | | | π/+/–/M+/% |
| 88TS | AL | 2 | 12 | 25 | | X | 1 | 8 | | X | X | X | | DIG | 7 + 2 | X | X | | | π/+/–/M+/rad |
| 108T | AL | 1 | 14 | 25 | | X | | | | X | X | X | | DIG | 10 + 2 | X | X | X | | π/+/–/M+/M–/rad |
| 120P | ARL | | | 29 | X | | 1 | 12 | X | X | | | X | | | | | X | Drucker | Tischrechner/% |
| 121C | ARL | | 1 | 27 | X | | 1 | 12 | X | X | | | X | DIG | 12 + 1 | | X | | | %/M+/M– |
| 121P | ARL | | 1 | 25 | X | | | | X | X | | | X | | | | | X | Drucker | Tischrechner/% |
| 121PD | ARL | | 1 | 29 | X | | 1 | 12 | X | X | | | X | DIG | 12 + 1 | | | X | Drucker | Tischrechner/% |
| 310P | ARL | | | 30 | X | | 1 | 10 | X | X | | | X | | | | | X | Drucker | Tischrechner/+/–/M+/M– |
| 310PD | ARL | | | 30 | X | | 1 | 10 | X | X | | | X | DIG | 10 + 1 | | | X | wie 310P | wie 310P |
| 312PD | ARL | | | 30 | X | | 1 | 12 | X | X | | | X | DIG | 12 + 1 | | | X | wie 310P | wie 310P |
| 807 | ARL | | | 20 | X | | | | X | X | | | X | DIG | 8 + 1 | | | X | | Tischrechner/% |
| 816AD | AL | | 1 | 28 | X | | 1 | 8 | X | X | | | | DIG | 8 + 1 | X | | X | | Tischrechner/%/M+/M– |
| 1205P | ARL | | | 26 | X | | | | X | X | | | X | | | | | X | Drucker | Tischrechner/% |
| 1206 | ARL | | | 23 | X | | | | X | X | | | X | DIG | 12 + 1 | | | X | | Tischrechner/%/+/– |
| 1215P | ARL | | | 32 | X | | 1 | | X | X | | | X | | | | | | wie 1205P | wie 1205P |
| 1216 | ARL | | | 23 | X | | | | X | X | | | X | DIG | 12 + 1 | | | X | | wie 1206 |
| 1216AD | ARL | | | 23 | X | | | | X | X | | | X | DIG | 12 + 1 | X | | X | | wie 1206 |
| 1217 | ARL | | | 22 | X | | 1 | 8 | X | X | | | X | DIG | 8 + 1 | | | X | | Tischrechner/%/M+/M– |
| 1218 | ARL | | 1 | 26 | X | | 1 | 12 | X | X | | | X | DIG | 12 + 1 | | | X | | wie 1217 |
| 1228PD | ARL | | 1 | 32 | X | | 2 | 12 | X | X | | | X | DIG | 12 + 1 | | | X | Drucker | Tischrechner/%/+/–/M+/M– |
| 1425P | ARL | | 2 | 37 | X | | 2 | 14 | X | X | | | X | | | | | X | Drucker (14 Stellen) | Tischrechner/%/+/–/M+/M– |
| 1977 | AL | | 1 | 24 | X | | 1 | 8 | | X | | | | LCD | 8 + 1 | | X | | | %/+/–/M+/M–/1600 Stunden |
| D100 | ARL | | | 25 | X | | 1 | 10 | X | X | | | X | DIG | 10 + 1 | X | X | X | | kleiner Tischrechner/%/Δ %/M+/M– |
| L800 | AL | | 1 | 24 | X | | 1 | 8 | | X | | | | LCD | 8 + 1 | | X | | | %/+/–/M+/M–/1500 Stunden |
| L811 | AL | | 1 | 24 | X | | 1 | 8 | | X | | | | LCD | 8 + 1 | | X | | | π/%/+/–/M+/M–/1500 Stunden |
| L812 | AL | | 4 | 28 | X | | 1 | 8 | | X | | | | LCD | 8 + 1 | | X | | | π/%/+/–/M+/M–/1500 Stunden |
| L813 | AL | | 4 | 27 | X | | 1 | 8 | X | X | | | | LCD | 8 + 1 | | X | | | wie L812 |
| L1019 | ARL | | | 25 | X | | 1 | 10 | X | X | | | | LCD | 10 + 1 | | X | | | kleiner Tischrechner/%/+/–/M+/M– |
| LC110 | AL | | | 26 | X | | 1 | 10 | X | X | | | | LCD | 10 + 1 | | X | | | wie L1019 |
| LM-8 | AL | | | 23 | X | | 1 | 8 | | X | | | | LCD | 8 + 1 | | X | | | %/M+/M– |
| LS822 | AL | 2 | 16 | 35 | | X | 1 | 8 | | | X | X | | LCD | 7 + 2 | | X | | | π/+/–/M+/M– |
| LS823 | AL | 2 | 16 | 41 | X | | 1 | 8 | | | X | X | | LCD | 7 + 2 | | X | | | π/+/–/M+/Statistik/Hierarchie |
| LS1002 | AL | 2 | ca. 30 | 37 | | X | 1 | 10 | | X | X | X | | LCD | 9 + 2 | | X | | | π/+/–/M+/rad/Hyp./Statistik |
| LT9 | AL | | | 22 | | X | 1 | 8 | | X | | | | LCD | 8 + 1 | | X | | | %/M+/M–/Uhr/Wecker |
| LT831 | AL | | 1 | 31 | X | | | | | X | | | | LCD | 8 + 1 | | X | | | %/+/–/M+/M–/Uhr/Wecker/Datum |
| Lady | AL | | | 19 | X | | | | | X | | | | DIG | 8 + 1 | | X | | | % |
| PD1 | AL | | | 28 | X | | 1 | 10 | X | X | | | | DIG | 10 + 1 | X | | X | Drucker | kleiner Tischrechner/%/+/–/M+/M– |

| Hersteller Typ | Rechenlogik | Anzahl der Klammern | Anzahl der Funktionen | Anzahl der Tasten | Tastenbeleg. einfach | Tastenbeleg. mehrfach | Zahl der Speicher | Zahl der Stellen | Komma Fest- | Komma Fließ- | Komma Gleit- | techn.-wiss. | kaufmännisch | Anzeige Art | Anzeige Stellen | Stromversorgung Akku | Stromversorgung Batterie | Stromversorgung Netz | Ergänzungen | Besonderheiten |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| noch | | | | | | | | | | | | | | | | | | | | |
| • **TRIUMPH ADLER** | | | | | | | | | | | | | | | | | | | | |
| SOLAR1981 | AL | | 4 | 27 | X | | 1 | 8 | X | X | | | | LCD | 8 + 1 | | | | | $\pi$/%/$^+/_-$/M+/M−/Solarzellen |
| SOLAR-MINI | AL | | 1 | 24 | X | | 1 | 8 | | X | | | | LCD | 8 + 1 | | | | | %/M+/M−/Solarzellen |
| • **VICTOR** | | | | | | | | | | | | | | | | | | | | |
| 100 | ARL | | | 23 | X | | | | | X | | | X | | | | | X | Drucker (10 Stellen) | kleiner Tischrechner/% |
| 116 | AL | | 1 | 28 | X | | 1 | 12 | X | X | | | | LCD | 12 + 1 | | X | | | kleiner Tischrechner/%/M+ |
| 204 | ARL | | | 27 | X | | 1 | 12 | X | X | | | X | DIG | 12 + 1 | | | X | | wie 116 |
| 210 | ARL | | 1 | 30 | X | | 2 | 14 | X | X | | | X | DIG | 14 + 1 | | | X | | wie 204 |
| 605 | ARL | | | 23 | X | | | | X | X | | | X | | | | | X | Drucker (12 Stellen) | Tischrechner/% |
| 610 | ARL | | | 26 | X | | | | X | X | | | X | | | | | X | wie 605 | wie 605 |
| 660 | ARL | | | 31 | X | | 1 | 12 | X | X | | | X | | | | | X | wie 605 | wie 605/M+ |
| 670 | ARL | | | 31 | X | | 1 | 12 | X | X | | | X | DIG | 12 + 1 | | | X | wie 660 | wie 660 |
| MEC/225 | AL | | | 24 | X | | 1 | 12 | X | X | | | | LED | 12 + 1 | X | | X | | %/$^+/_-$/M+/M− |
| MINI | ARL | | | 17 | X | | | | X | X | | | | LED | 8 + 1 | X | | | | |
| • **WALTHER** | | | | | | | | | | | | | | | | | | | | |
| ETR1090 | AL | 2 | 14 | 37 | X | | 1 | 8 | | X | X | X | | DIG | 8 + 2 | X | | X | | Tischrechner/$\pi$/$^+/_-$/M+/rad |

## 1.2 Programmierbare Taschen- und Tischrechner

Die in den folgenden Datentabellen aufgelisteten programmierbaren Taschen- und Tischrechner sind ebenso wie die nicht-programmierbaren nach Herstellern alphabetisch geordnet. Die in der Spalte „Rechenlogik" verwendeten Abkürzungen sind die gleichen wie bei den nicht-programmierbaren Rechnern, also:

AL – Algebraische Logik
AOS – Algebraisches Operationssystem
ARL – Arithmetische Logik
UPN – Umgekehrte Polnische Notation.

Unterschiede zu den Datentabellen nicht-programmierbarer Rechner sind nur darin zu sehen, daß hier die Anzahl der möglichen „Programmschritte" angegeben ist. Zusätzlich wird gesagt, wieviele Tastenbetätigungen einen Programmschritt ausmachen („Tasten pro Schritt"). Die Regel ist dabei, daß jeder Tastendruck einem Programmschritt entspricht (1 Taste/Schritt). Seltener werden mehrere Tastenbetätigungen zu einem Schritt zusammengefaßt (günstigere Nutzung des Programmspeichers).

| Hersteller Typ | Rechenlogik | Programmschritte | Tasten/Schritt | Anzahl der Klammern | Anzahl der Funktionen | Anzahl der Tasten | Zahl der Speicher | Zahl der Stellen | Komma Fest- | Komma Fließ- | Komma Gleit- | techn.-wiss. | kaufmännisch | Anzeige Art | Anzeige Stellen | Stromversorgung Akku | Stromversorgung Batterie | Stromversorgung Netz | Ergänzungen | Besonderheiten |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Canon** | | | | | | | | | | | | | | | | | | | | |
| SV-10 | ARL | | | | | 41 | 16 | 12 | X | X | | | X | DIG | 12 + 3 | | | X | Drucker | Tischrechner/%/+/_/Statistik/über Einschub-EPROM programmierbar |
| SX-100 | AL | 200 (1 000) | 1 | 2 | 16 | 74 | 20 (100) | 14 | | | X | X | X | LED | 16 + 2 | | | X | Drucker/Magnetkartenleser | Tischrechner/π/e/rad |
| SX-300 | AL | 500 (4 000) | 1 | 2 | 16 | 74 | 30 (500) | 14 | | | X | X | X | LED | 16 + 2 | | | X | Drucker/Kassetten-Recorder, Schreibmaschine, Lochstreifenstanzer, -leser | Tischrechner/π/e/rad |
| **Casio** | | | | | | | | | | | | | | | | | | | | |
| fx-1 | AL | 127 | 1 | | 12 | 43 | 11 | 10 | | | X | X | | DIG | 12 + 2 | | X | X | Magnetkartenleser | π/M+/M–/+/_/rad/Hexadezimal-Umrechnung |
| fx-201P | AL | 127 | 1 | | 12 | 43 | 11 | 8 | | | X | X | | DIG | 10 + 2 | | X | X | | π/M+/M–/+/_/rad/Hexadezimal-Umrechnung |
| fx-202P | AL | 127 | 1 | | 12 | 43 | 11 | 8 | | | X | X | | DIG | 10 + 2 | | X | X | | wie fx-201/zusätzlich: Permanentspeicher (1 Jahr) |
| fx-501P | AL | 128 | > 1 | 5 | 30 | 50 | 11 | 10 | | | X | X | | LCD | 12 + 2 | | X | | Recorderanschluß | π/%/+/_/M+/M–/Statistik/Permanentspeicher |
| fx-502P | AL | 256 | > 1 | 10 | 30 | 50 | 22 | 10 | | | X | X | | LCD | 12 + 2 | | X | | wie fx-501P | wie fx-501P |
| PRO-101 | AL | 256 | 1 | | | 38 | 16 | 10 | X | X | | | | DIG | 10 + 1 | | X | X | | %/M+/M–/+/_/Permanentspeicher |
| **Citizen** | | | | | | | | | | | | | | | | | | | | |
| 140PR | ARL | | | | | 40 | 1 | 12 | X | X | | | X | DIG | 12 + 1 | | | X | Drucker | Tischrechner/%/Δ%/+/_/M+/M– |
| **Commodore** | | | | | | | | | | | | | | | | | | | | |
| P50 | AL | 24 | 1 | | 14 | 49 | 1 | 8 | | | X | X | | LED | 10 + 2 | | X | X | | π/+/_/M+/Mx/rad |
| PR100 | AL | 72 | 1 | 4 | 20 | 45 | 10 | 8 | | | X | X | | LED | 10 + 2 | X | | X | | M+/M–/rad/π/%/Δ%/+/_/Statistik/Umwandlungen |
| **Compucorp** | | | | | | | | | | | | | | | | | | | | |
| 322G | AL | 80 | 1 | 2 | 17 | 43 | 10 | 13 | X | | X | X | | LED | 12 + 2 | X | | | | π/+/_/rad |
| 324G | AL | 160 | 1 | 2 | 17 | 43 | 10 | 13 | X | | X | X | | LED | 12 + 2 | X | | | | π/+/_/rad |
| 325 | AL | 416 | 1 | 4 | 20 | 52 | 12 (140) | 13 | X | | X | X | | LED | 12 + 2 | | | X | Drucker/Magnetband-Kassette/Sichtgerät | π/+/_/rad/Statistik/Umrechnungen |
| 326 | AL | 160 | 1 | 4 | 20 | 46 | 12 | 13 | X | | X | X | | LED | 10 + 2 | X | | X | Magnetband-Kassette, Sichtgerät | wie 325 |
| 327 | AL | 416 | 1 | 4 | 20 | 52 | 40 | 13 | X | | X | X | | LED | 12 + 2 | | | X | wie 325 | wie 325 |
| 344 | AL | 160 | 1 | 2 | 15 | 43 | 10 | 13 | X | | X | X | X | LED | 12 + 2 | X | | | | +/_/Umrechnungen/Statistik-Rechner |
| 425 | AL | 512 (2 048) | 1 | | | | 256 | 13 | X | | X | X | | | | | | X | Drucker, vollständige Peripherie | |
| 445 | AL | 2 048 | 1 | | | | 256 | 13 | X | | X | X | | | | | | | wie 425 | Statistik-Rechner |
| 481 | AL | 2 048 | 1 | | | | 256 | 13 | X | | X | | X | | | | | | wie 425 | kaufmännische |
| 485 | AL | 2 048 | 1 | | | | 256 | 13 | X | | X | | X | | | | | | wie 425 | Rechner |
| **Hewlett Packard** | | | | | | | | | | | | | | | | | | | | |
| HP-19C | UPN | 98 | ≤ 3 | | 16 | 31 | 30 | 10 | X | | X | X | | LED | 10 + 2 | X | | | Drucker | π/%/M+/M–/Statistik/16 Permanentspeicher |
| HP-25 | UPN | 49 | ≤ 3 | | 16 | 30 | 8 | 10 | X | | X | X | | LED | 10 + 2 | X | | | | π/%/M+/M–/Statistik |
| HP-25C | UPN | 49 | ≤ 3 | | 16 | 30 | 8 | 10 | X | | X | X | | LED | 10 + 2 | X | | | | π/%/M+/M–/Statistik |
| HP-29C | UPN | 98 | ≤ 3 | | 16 | 31 | 30 | 10 | X | | X | X | | LED | 10 + 2 | X | | | | π/%/M+/M–/Statistik/16 Permanentspeicher |
| HP-33E | UPN | 49 | ≤ 3 | | 18 | 30 | 8 | 10 | X | | X | X | | LED | 10 + 1 | X | | | | %/Δ%/π/rad/Statistik |
| HP-38E | UPN | 99 | ≤ 3 | | 10 | 30 | 25 | 10 | X | | X | | X | LED | 10 + 1 | X | | | | %/Δ%/Zins/Statistik/Kalender |
| HP-55 | UPN | 49 | ≤ 3 | | 19 | 35 | 20 | 10 | X | | X | X | | LED | 12 + 2 | X | | | | %/Δ%/π/rad/Polar/Umwandlungen/Stoppuhr/Statistik |
| HP-65 | UPN | 100 | ≤ 3 | | 9 | 35 | 14 | 10 | X | | X | X | | LED | 12 + 2 | X | | | Magnetstreifen | π/rad/Polar/Oktal/Dezimal |
| HP-67 | UPN | 224 | ≤ 3 | | 17 | 35 | 26 | 10 | X | | X | X | | LED | 12 + 2 | X | | X | Magnetstreifen | π/%/Δ%/rad/Polar/Statistik |

| Hersteller Typ | Rechenlogik | Programmschritte | Tasten/Schritt | Anzahl der Klammern | Anzahl der Funktionen | Anzahl der Tasten | Zahl der Speicher | Zahl der Stellen | Komma Fest- | Komma Fließ- | Komma Gleit- | techn.-wiss. | kaufmännisch | Anzeige Art | Anzeige Stellen | Stromversorgung Akku | Stromversorgung Batterie | Stromversorgung Netz | Ergänzungen | Besonderheiten |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| noch | | | | | | | | | | | | | | | | | | | | |
| ● **Hewlett Packard** | | | | | | | | | | | | | | | | | | | | |
| HP-97 | UPN | 224 | ≤ 3 | | 17 | 56 | 26 | 10 | X | | X | X | | LED | 12 + 2 | X | | X | Drucker, Magnetstreifen | Tischrechner/π/%/Δ %/rad/Polar/Statistik |
| HP-97S I/O | UPN | 224 | ≤ 3 | | 17 | 56 | 26 | 10 | X | | X | X | | LED | 12 + 2 | X | | X | wie HP-97 zusätzlich: | wie HP-97 Peripherieanschluß |
| ● **Litronix** | | | | | | | | | | | | | | | | | | | | |
| 2290 | ARL | 10 | 1 | | | 25 | 1 | 8 | | X | | | | LED | 8 + 1 | | X | X | | %/M+/M– |
| ● **MBO** | | | | | | | | | | | | | | | | | | | | |
| TRS1300P | ARL | 14 | 1 | | 1 | 30 | 2 | 12 | X | X | | | X | DIG | 12 + 1 | | | X | Drucker | Tischrechner/%/+/– |
| TRS1350PD | ARL | 14 | 1 | | 1 | 34 | 2 | 12 | X | X | | | X | DIG | 12 + 1 | | | X | Drucker | Tischrechner/%/+/–/M+/M– |
| ● **National** | | | | | | | | | | | | | | | | | | | | |
| Novus 4515 | UPN | 100 | 1 | | 13 | 32 | 1 | 8 | | X | | X | | LED | 8 + 1 | X | | X | | wie Novus 4510 |
| Novus 4525 | UPN | 100 | 1 | | 13 | 35 | 1 | 8 | | X | X | X | | LED | 10 + 2 | X | | X | | wie Novus 4520 |
| Novus 6025 | ARL | 100 | 1 | | 10 | 32 | 1 | 8 | | X | | | X | LED | 8 + 1 | X | | X | | wie Novus 6020 |
| Novus 6035 | ARL | 100 | 1 | | 10 | 32 | 1 | 8 | | X | | X | | LED | 8 + 1 | X | | X | | wie Novus 6030 |
| ● **NIKKO** | | | | | | | | | | | | | | | | | | | | |
| 120PPD | ARL | 256 | 1 | | 1 | 34 | 1 | 12 | X | X | | | X | DIG | 12 + 1 | | | X | Drucker | Tischrechner/%/+/–/M+/M– |
| ● **Quelle-Privileg** | | | | | | | | | | | | | | | | | | | | |
| PR-56D-NC | AL | 72 | 1 | 5 | 14 | 40 | 10 | 8 | | X | X | X | | DIG | 10 + 2 | X | | X | | π/+/–/M+/M–/M x/M ÷ |
| ● **Rockwell** | | | | | | | | | | | | | | | | | | | | |
| 475P/D | ARL | 16 | 1 | | | 35 | 1 | 12 | X | X | | | X | DIG | 12 + 1 | | | X | Drucker | Tischrechner/%/M+/M– |
| 920-1 | AL | 192 | ≤ 2 | 3 | 5 | 53 | 9 | 12 | X | X | X | X | | LED | 12 + 2 | | | X | Drucker/Magnetkartenleser, Lochstreifenleser | Tischrechner/%/+/–/M+/M– |
| 920-2 | AL | 448 | ≤ 2 | 3 | 5 | 53 | 41 | 12 | X | X | X | X | | LED | 12 + 2 | | | X | wie 920-1 | wie 920-1 |
| 920-3 | AL | 960 | ≤ 2 | 3 | 5 | 53 | 99 | 12 | X | X | X | X | | LED | 12 + 2 | | | X | wie 920-1 | wie 920-1 |
| 930-2 | AL | 448 | ≤ 2 | 3 | 5 | 64 | 41 | 12 | X | X | X | X | | LED | 12 + 2 | | | X | wie 920-1 | wie 920-1/zusätzlich 11 frei belegbare Tasten |
| 930-3 | AL | 960 | ≤ 2 | 3 | 5 | 64 | 99 | 12 | X | X | X | X | | LED | 12 + 2 | | | X | wie 920-1 | wie 930-2 |
| 940-2 | AL | 448 | ≤ 2 | 3 | 16 | 64 | 41 | 12 | X | X | X | | X | LED | 12 + 2 | | | X | wie 920-1 | wie 920-1/zusätzlich 11 finanztechnische Funktionen |
| 960-1 | AL | 256 | ≤ 2 | 3 | 22 | 64 | 28 | 12 | X | X | X | X | | LED | 12 + 2 | | | X | wie 920-1 | π/+/–/rad/Polar/Statistik/Umrechnungen/10 Tasten frei verfügbar |
| 960-2 | AL | 512 | ≤ 2 | 3 | 22 | 64 | 60 | 12 | X | X | X | X | | LED | 12 + 2 | | | X | wie 920-1 | wie 960-1 |
| 960-3 | AL | 1 000 | ≤ 2 | 3 | 22 | 64 | 92 | 12 | X | X | X | X | | LED | 12 + 2 | | | X | wie 920-1 | wie 960-1 |
| ● **Sanyo** | | | | | | | | | | | | | | | | | | | | |
| CZ-0911PG | AL | | 1 | 5 | 14 | 40 | 10 | 10 | X | X | X | X | | DIG | 10 + 2 | X | X | X | | π/+/–/M+/M–/M x/M ÷/rad |
| EL-5100 | AL | 400 | 1 | 15 | 40 | 60 | 10 | 10 | | X | X | X | | LCD | 24 | | X | | | π/+/–/M+/Statistik/alphanumerische Eingabe bis 80 Stellen/Permanentspeicher |
| EL-5101 | AL | 80 | 1 | 15 | 40 | 55 | 5 | 10 | | X | X | X | | LCD | 16 | | X | | | wie EL-5100 |
| ICC-163PS | ARL | 62 | 1 | | 1 | 33 | 2 | 16 | X | X | | | X | | | | X | | Drucker (16 Stellen) | Tischrechner/%/M+/M– |
| PC-1201 | AL | 128 | 1 | 1 | 22 | 40 | 1 | 10 | | X | X | X | | LED | 10 + 2 | X | | | | π/+/–/M+/Statistik |
| PC-1300S | AL | 640 | 1 | | | 56 | 42 | 16 | X | X | X | X | | LED | 16 | X | | | Drucker/Magnetstreifen | volle Ausstattung |
| ● **SHARP** | | | | | | | | | | | | | | | | | | | | |
| CS-364P-III | ARL | 432 | 1 | | 1 | 42 | 12 | 16 | X | X | | | X | DIG | 16 + 1 | | | X | Drucker/Magnetkarten | Tischrechner/M+ |
| CS-365P | ARL | 476 | 1 | | 1 | 46 | 24 | 16 | X | X | | | X | DIG | 16 + 1 | | | X | wie 364P-III | wie 364P-III |
| CS-365PT | AL | 576 | 1 | | | 46 | 24 | 16 | X | X | | X | | DIG | 16 + 1 | | | | wie 364P-III/Schreibmaschine | wie 364P-III/techn.-wiss. |
| CS-762R | ARL | 72 | 1 | | | 40 | 3 | 16 | X | X | | | X | DIG | 16 + 1 | | | X | Drucker | Tischrechner/%/M+/M– |

| Hersteller Typ | Rechenlogik | Programmschritte | Tasten/Schritt | Anzahl der Klammern | Anzahl der Funktionen | Anzahl der Tasten | Zahl der Speicher | Zahl der Stellen | Komma Fest- | Komma Fließ- | Komma Gleit- | techn.-wiss. | kaufmännisch | Anzeige Art | Anzeige Stellen | Stromversorgung Akku | Stromversorgung Batterie | Stromversorgung Netz | Ergänzungen | Besonderheiten |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| noch | | | | | | | | | | | | | | | | | | | | |
| ● **SHARP** | | | | | | | | | | | | | | | | | | | | |
| EL-5100 | AL | 400 | | 8 | 40 | 60 | | | | | X | X | | LCD | 24 | | X | | | $\pi$/rad/Polar/Permanent-Speicher |
| PC-1001 | AL | 64 | 1 | | 14 | 31 | 8 | 10 | | X | X | X | | DIG | 12 + 2 | | | X | | Tischrechner/$\pi$/M+/M– |
| PC-1002 | AL | 64 | 1 | | 16 | 31 | 8 | 10 | | X | X | X | | DIG | 12 + 2 | | | X | | Tischrechner/$^{+}/_{-}$/$\pi$/M+/M–/rad/Polar |
| PC-2600 | AL | 512 | 1 | | | 55 | 28 | 10 | | X | | X | | DIG | 10 + 1 | | | X | Drucker/Schreibmaschine/Lochstreifen | Tischrechner |
| ● **Texas Instruments** | | | | | | | | | | | | | | | | | | | | |
| SR-52 | AOS | 224 | 1 | 9 | 16 | 45 | 20 | 12 | X | | X | X | | LED | 12 + 2 | X | | | Magnetstreifen, Druckeranschluß | $\pi$/M+/M–/M x/M ÷/rad/Polar |
| SR-56 | AOS | 100 | 1 | 9 | 20 | 40 | 10 | 12 | X | | X | X | | LED | 12 + 2 | X | | | Druckeranschluß | $\pi$/M+/M–/M x/M ÷/Polar/Statistik |
| SR-60 | AOS | 480 | 1 | 9 | 24 | 95 | 40 | 12 | X | | X | X | | LED | 12 + 2 | | | X | Magnetstreifen, Drucker | Tischrechner/$\pi$/%/$\Delta$ %/rad/Polar/M+/M–/M x/M ÷ |
| TI-42MBA | AOS | 32 | 1 | 12 | 25 | 40 | 12 | 11 | X | X | | | X | LED | 10 + 1 | X | | | | %/$\Delta$ %/Statistik/Finanztechnik |
| TI-51-III | AOS | 32 | 1 | 9 | 20 | 40 | 10 | 11 | | | X | X | | LED | 10 + 2 | X | | X | | $^{+}/_{-}$/$\pi$/M+/M–/rad/Polar/Umwandlungen |
| TI-53 | AOS | | | 15 | 15 | 40 | 1 | 11 | | | X | X | | LCD | 7 + 2 | | X | | | $\pi$/$^{+}/_{-}$/M+/Permanentspeicher/1000 Stunden |
| TI-57 | AOS | 50 | ⩽ 3 | 9 | 9 | 40 | 8 | 12 | X | | X | X | | LED | 10 + 2 | X | | | | $\pi$/M+/M–/M x/M ÷/rad/Polar |
| TI-58 | AOS | 480 | | 9 | 9 | 45 | 60 | 12 | X | | X | X | | LED | 10 + 2 | X | | | ROM, Druckeranschluß | wie TI-57 |
| TI-59 | AOS | 960 | | 9 | 9 | 45 | 100 | 12 | X | | X | X | | LED | 10 + 2 | X | | | ROM, Magnetstreifen, Druckeranschluß | wie TI-57 |
| TI-5230 | ARL | 200 | 1 | | | 42 | 1 | 12 | X | X | | | X | DIG | 12 + 1 | | | X | Drucker | %/M+/M–/Permanentspeicher/Textanzeige |
| ● **VICTOR** | | | | | | | | | | | | | | | | | | | | |
| 4900A | ARL | 1792 | 1 | | | 49 | | | X | X | | | X | DIG | 14 + 1 | | | X | Drucker/Magnetkarten | Tischrechner/alphanumerisch |
| 4950 | ARL | 1792 | 1 | | | 49 | | | X | X | | | X | DIG | 14 + 1 | | | X | zusätzlich Kassette | wie 4900 A |

## 1.3 Mikroprozessoren

Für die Datentabellen „Mikroprozessoren" wurde folgende Systematik verwendet:

- Es sind getrennte Tabellen aufgestellt für 1bit-$\mu$P, 2bit-$\mu$P, 4bit-$\mu$P, 8bit-$\mu$P, 12bit-$\mu$P und 16bit-$\mu$P;
- Innerhalb jeder Prozessorgruppe wurde nach Prozessortypen sortiert.

Diese Gliederung entspricht der Konvention für die Einteilung und Bewertung von Mikroprozessoren. Es wird nämlich eher nach Daten für den 8bit-Prozessor 8085 gesucht werden, als daß man sich dafür interessiert, welche verschiedenen Typen Intel oder Siemens herstellen. Die Sortierung wurde nach Möglichkeit so gewählt, daß zuerst ansteigende Typenzahlen angeordnet sind, also z. B. 2650 vor 6502 und erst später 6800, 8080 etc. Danach sind die mit Buchstaben beginnenden Typen alphabetisch geordnet.

In die Spalte „Technologie" ist eingetragen, nach welchem der bekannten Herstellungsverfahren der jeweilige Prozessor hergestellt wird oder wurde. Als „Maschinenzyklus" ist der Systemtakt in MHz eingetragen. Ferner ist angegeben, wieviele zeitlich gegeneinander versetzte Taktsteuerungen nötig sind und ob sie intern erzeugt werden oder von außen (extern) eingespeist werden müssen.

Ein wichtiger Hinweis muß gegeben werden: Wenn eine Spalte nicht ausgefüllt ist, bedeutet das in der Regel, daß die betreffende Eigenschaft nicht vorhanden ist. Aber leider kann dies nicht als durchgehend verbindlich angesehen werden, denn nicht immer ist aus technischen Unterlagen jede Einzelheit klar zu entnehmen. Es kann ein fehlender Eintrag also auch bedeuten, daß sie trotzdem vorhanden ist, aber nicht ermittelt werden konnte.

Die letzte Spalte „Hersteller" nennt zuerst immer den „Original"-Hersteller. Danach sind, sofern bekannt, die Zweithersteller (*Second Sources*) angegeben.

**1bit-$\mu$P**

| Typ | Technologie | Maschinenzyklus (MHz) | Takt | | Befehlsausführung ($\mu$s) | | | Befehle | Adressierungsarten | Register | | | Stack | Adreßraum (Worte) | DMA | Ein-Ausgänge | Interrupts | Versorgung (V) | Pins | RAM | ROM | Besonderheiten | Hersteller |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | Phasen | ext./int. | kürzeste | längste | mittlere | | | Akkus | Index | allgemein | | | | | | | | | | | |
| 14500 | CMOS | 1 | | int. | 1 | 1 | 1 | 16 | | | | 1 | 0 | 0 | | | 1 | 3–18 | 16 | | | Serielles Steuerwerk | Motorola |
| SBA | NMOS | 0,8 | | int. | | 1280 | | 8 | | | | RAM | RAM | | | 30 | | 5,12 | 40 | 120bit | 1K byte | | General Instruments |

**2bit-$\mu$P**

| Typ | Technologie | Maschinenzyklus (MHz) | Takt | | Befehlsausführung ($\mu$s) | | | Befehle | Adressierungsarten | Register | | | Stack | Adreßraum (Worte) | DMA | Ein-Ausgänge | Interrupts | Versorgung (V) | Pins | RAM | ROM | Besonderheiten | Hersteller |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | Phasen | ext./int. | kürzeste | längste | mittlere | | | Akkus | Index | allgemein | | | | | | | | | | | |
| 3001 | STTL | 6,6 | 1 | ext. | 0,06 | 0,15 | 0,10 | | | | | | | | | | | 5 | 40 | | | Bis 320bit kaskadierbar (Mikroprogramm-Steuerschaltung) | Intel, Valvo-Signetics |
| 3002 | STTL | 6,6 | 1 | ext. | 0,06 | 0,15 | 0,10 | 8 | | 1 | | 10 | | | | | | 5 | 28 | | | Bis 320bit kaskadierbar (Prozessorelement) | " |

| Typ | Technologie | Maschinen-zyklus (MHz) | Takt | | Befehlsausführung (μs) | | | Befehle | Adressierungsarten | Register | | | Stack | Adreßraum (Worte) | DMA | Ein-Ausgänge | Interrupts | Versorgung (V) | Pins | RAM | ROM | Besonderheiten | Hersteller |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | Phasen | ext./int. | kürzeste | längste | mittlere | | | Akkus | Index | allgemein | | | | | | | | | | | |
| 402 | NMOS | 4 | | int. | | | 4 | 57 | | 1 | | | 3 x 10 | | | 23 | | 5 | 40 | 64 x 4 | | ADC enthalten | National, Western Digital |
| 411L | NMOS | 4 | | int. | | | 16 | 43 | | 1 | | | 3 x 10 | | | 16 | | 5 | 20 | 32 x 4 | 512 x 8 | Einchipfamilie COPS-II | " |
| 420 | NMOS | 4 | | int. | | | 4 | 57 | | 1 | | | 3 x 10 | | | 23 | | 5 | 28 | 64 x 4 | 1024 x 8 | " | " |
| 420C | CMOS | 4 | | int. | | | 16 | 57 | | 1 | | | 3 x 10 | | | 23 | | 5 | 28 | 64 x 4 | 1024 x 8 | " | " |
| 440 | NMOS | 4 | | int. | | | 4 | | | 1 | | | 3 x 10 | | | 35 | | 5 | 40 | 128 x 4 | 2048 x 8 | ADC enthalten | |
| 1872 | PMOS | 0,15 | | int. | 6,25 | 12,5 | | 37 | | | | RAM | | | | 27 | 1 | 12 | 40 | 4 x 32 | 512 x 10 | keine Speicher-erweiterung möglich, Ein-Chip-μP | Western Digital |
| 2272 | PMOS | 0,15 | | int. | 6,25 | 12,5 | | 37 | | | | RAM | | | | 27 | 1 | 12 | 40 | | | Speicher wie 1872, andere Ein-Ausgaben | " |
| 2901 | LPSTTL | 12,5 | | | 0,06 | | | 16 | | | | 16 | | | | | | 5 | 40 | 16 x 4 | | kaskadierbar | AMD, Fairchild, Motorola, NEC, Raytheon, Sescosem, Siemens, Valvo-Signetics |
| 2903 | LPSTTL | 12,5 | | | 0,06 | | | 25 | | | | 16 | | | | | | 5 | 48 | | | kaskadierbar | wie 2901 |
| 4004 | PMOS | 0,74 | 2 | ext. | 10,8 | 21,6 | | 46 | | | | 16 | 3 x 12 | 4K | | | 1 | 15 | 16 | | | | Intel, National |
| 4040 | PMOS | 0,74 | 2 | ext. | 10,8 | 21,6 | | 60 | | | | 24 | 7 x 12 | 8K | | | 1 | 15 | 24 | | | Standard-4bit-Prozessor | Intel, Siemens |
| 5701 | STTL | 10 | | ext. | | | 0,1 | 32 | | | | 16 | | | | | | 5 | 40 | | | kaskadierbar, militärische Version | MMI |
| 6701 | STTL | 10 | | ext. | | | 0,1 | 36 | | | | 16 | | | | | | 5 | 40 | | | kaskadierbar | MMI |
| 10800 | ECL | 20 | | | | | 0,05 | | | | | | | | | | | –5 | 48 | | | kaskadierbar, sehr schnell | Motorola, Fairchild |
| 92901 | LPSTTL | 9,5 | | | 0,1 | | | 14 | | | | | | | | | | 5 | 40 | 16 x 4 | | kaskadierbar | Thomson-CSF |
| 141000 | CMOS | 0,7 | | int. | | | | 43 | | | | | | | | 23 | | 3–6 | 28 | 64 x 4 | 1K byte | Ein-Chip-μP | Motorola, Texas |
| 141200 | CMOS | 0,7 | | int. | | | | 43 | | | | | | | | 28 | | 3–6 | 40 | | | Ein-Chip-μP, wie 141000 | Motorola |
| 74S481 | STTL | 11 | | ext. | | | 0,1 | | | | | 2 | | | | | | 5 | 48 | | | kaskadierbar | Texas |
| 74LS481 | LPSTTL | 8,3 | | ext. | | | 0,1 | | | | | 2 | | | | | | 5 | 48 | | | kaskadierbar | Instruments |
| 54S481 | STTL | 11 | | ext. | | | 0,1 | | | | | 2 | | | | | | 5 | 48 | | | militärische | " |
| 54LS481 | LPSTTL | 8,3 | | ext. | | | 0,1 | | | | | 2 | | | | | | 5 | 48 | | | Version | " |
| AM2S01A | | | | | | | | | | | | | | | | | | | | | | siehe 2901 | |
| AM2903 | | | | | | | | | | | | | | | | | | | | | | siehe 2903 | |
| e1159 | CMOS | 0,7 | | int. | | | 240 | 9 | | | | | | | | 6 | | 1,55 | 64 | 32 x 4 | | Ein-Chip-μP, Uhrensteuerung | Eurosil |
| IMP-4 | PMOS | | | | | | 11,2 | 42 | | | | 4 | 7 x 12 | 4K | | | X | +5 –12 | 24 | | | | National |
| Macrologic 9405A | STTL | 13 | | ext. | | | 0,1 | | | | | | | | | | | 5 | | | | kaskadierbar | Fairchild, Signetics |
| 9405B | CMOS | 2 | | ext. | | | | | | | | | | | | | | | | | | " | " |

| Typ | Technologie | Maschinen-zyklus (MHz) | Takt | | Befehlsausführung (μs) | | | Befehle | Adressierungsarten | Register | | | Stack | Adreßraum (Worte) | DMA | Ein-Ausgänge | Interrupts | Versorgung (V) | Pins | RAM | ROM | Besonderheiten | Hersteller |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | Phasen | ext./int. | kürzeste | längste | mittlere | | | Akkus | Index | allgemein | | | | | | | | | | | |
| MM75 | PMOS | 0,1 | | int. | 10 | 40 | 12,5 | 68 | | | | 1 | | | | 22 | 1 | −15/ +5/−10 | 28 | 48 x 4 | 640 x 8 | Einchipfamilie PPS-4/1 | Rockwell, Telefunken |
| MM76 | PMOS | 0,1 | | int. | 10 | 40 | 12,5 | 69 | | | | 1 | | | | 31 | 1 | −15/ +5/−10 | 42 | 48 x 4 | 640 x 8 | " | " |
| MM76C | PMOS | 0,1 | | int. | 10 | 30 | | 69 | | | | 1 | | | | 39 | 1 | −15/ +5/−10 | 52 | 48 x 4 | 640 x 8 | " | " |
| MM76E | PMOS | 0,1 | | int. | 10 | 30 | | 69 | | | | 1 | | | | 31 | 1 | −15/ +5/−10 | 42 | 48 x 4 | 1024 x 8 | " | " |
| MM76EL | PMOS | 0,1 | | int. | 10 | 30 | | 69 | | | | 1 | | | | 31 | 1 | −15 +5 −10 | 40 | 48 x 4 | 1024 x 8 | " | " |
| MM76L | PMOS | 0,1 | | int. | 10 | 30 | | 69 | | | | 1 | | | | 31 | 1 | 6−11 | 40 | 48 x 4 | 640 x 8 | " | " |
| MM77 | PMOS | 0,1 | | int. | 10 | 40 | | 68 | | | | 2 | | | | 31 | 1 | −17 +5 −12 | 42 | 96 x 4 | 1344 x 8 | " | " |
| MM77L | PMOS | 0,1 | | int. | 10 | 40 | | 68 | | | | 2 | | | | 31 | 1 | −17 +5 −12 | 40 | 96 x 4 | 1536 x 8 | " | " |
| MM78 | PMOS | 0,1 | | int. | 10 | 40 | | 68 | | | | 2 | | | | 31 | 1 | −17 +5 −12 | 42 | 128 x 4 | 2048 x 8 | " | " |
| MN1400 | NMOS | 0,45 | | int. | | | 10 | 75 | | 1 | 2 | 1 | 2 | | | 30 | | 5 | 40 | 64 x 4 | 1024 x 8 | Ein-Chip-μP | Panasonic |
| MN1402 | NMOS | 0,45 | | int. | | | 10 | 57 | | 1 | 2 | 1 | 2 | | | 13 | | 5 | 28 | 32 x 4 | 768 x 8 | " | " |
| MN1498 | NMOS | 0,45 | | int. | | | 10 | 68 | | 1 | 2 | 1 | 2 | | | 17 | | 5 | 40 | 64 x 4 | | " | " |
| MN1499 | NMOS | 0,45 | | int. | | | 10 | 75 | | 1 | 2 | 1 | 2 | | | 25 | | 5 | 64 | 64 x 4 | | " | " |
| MN1450 | CMOS | | | int. | | | 10 | | | 1 | 2 | 1 | 2 | | | 28 | | | 40 | 64 x 4 | 1024 x 8 | " | " |
| μCOM-42 | PMOS | 0,2 | | int. | | | 10 | 72 | | | | | 4 | | | 35 | 2 | −10 | 42 | 94 x 4 | 1920 x 10 | Ein-Chip-μP nicht erweiterbar | NEC |
| μCOM-43 | PMOS | 0,4 | | int. | | | 10 | 80 | | | | | 3 | | | 35 | 1 | −10 | 42 | 94 x 4 | 2000 x 8 | " | " |
| μCOM-43 | CMOS | 0,4 | | int. | | | 10 | 80 | | | | | 3 | | | 35 | 1 | +5 | 42 | 94 x 4 | 2000 x 8 | " | " |
| μCOM-44 | PMOS | 0,4 | | int. | | | 10 | 58 | | | | | 1 | | | 35 | | −10 | 42 | 64 x 4 | 1000 x 8 | " | " |
| μCOM-44 | CMOS | 0,4 | | int. | | | 10 | 58 | | | | | 1 | | | 35 | | +5 | 42 | 64 x 4 | 1000 x 8 | " | " |
| μCOM-45 | PMOS | 0,4 | | int. | | | 10 | 58 | | | | | 1 | | | 21 | | −10 | 28 | 32 x 4 | 640 x 8 | Ein-Chip-μP nicht erweiterbar | NEC |
| μCOM-45 | CMOS | 0,4 | | int. | | | 10 | 58 | | | | | 1 | | | 21 | | +5 | 28 | 32 x 4 | 640 x 8 | " | " |
| μCOM-46 | PMOS | 0,4 | | int. | | | 10 | | | | | | 4 | | | 32 | 2 | −10 | 42 | 94 x 4 | 2000 x 8 | ADC enthalten | " |
| μPD541 | NMOS | 0,5 | | ext. | 6,4 | 38,4 | | 69 | | | | 4 | 8 x 12 | 4K | X | | 8 | +5 −5 | 42 | | | | NEC |
| μPD-545 | PMOS | | | | | | 10 | | | | | | | | | | | −10 | 42 | | | | " |
| μPD-546 | PMOS | | | | | | 10 | | | | | | | | | | | −10 | 42 | | | | " |
| PPS-4 | PMOS | 0,2 | 4 | ext. | 5 | 16 | | 50 | | 1 | 1 | 1 | | 8K | | 12 | 1 | −17 +5 −12 | 42 | | | | Rockwell, Telefunken |
| PPS-4/2 | PMOS | 0,2 | 4 | int. | 5 | 16 | | 50 | | 1 | 1 | 1 | | 8K | | 28 | | −17 +5 −12 | 42 | 128 x 4 | 2048 x 8 | Clock enthalten | " |

## 4bit-μP

| Typ | Technologie | Maschinen-zyklus (MHz) | Takt | | Befehlsausführung (μs) | | | Befehle | Adressierungsarten | Register | | | Stack | Adreßraum (Worte) | DMA | Ein-Ausgänge | Interrupts | Versorgung (V) | Pins | RAM | ROM | Besonderheiten | Hersteller |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | Phasen | ext./int. | kürzeste | längste | mittlere | | | Akkus | Index | allgemein | | | | | | | | | | | |
| S2000 | NMOS | 1 | | int. | | | 4 | 51 | | 1 | | | 3 x 10 | 8 K | | 29 | | 9 | 40 | 64 x 4 | 1024 x 8 | Tastatur-Eingang, Sieben-Segment-Ausgang | AMI |
| S2150 | NMOS | 1 | | int. | | | 4 | 51 | | 1 | | | 3 x 10 | 8 K | | 29 | | 9 | 40 | 80 x 4 | 1,5K byte | " | " |
| S2200 | NMOS | 1 | | int. | | | 4 | 59 | | 1 | | | 3 x 10 | 8 K | | 29 | | 9 | | 128 x 4 | 2K byte | " 8 bit-ADC | " |
| S2400 | NMOS | 1 | | int. | | | 4 | 59 | | 1 | | | 3 x 10 | 8 K | | 29 | | 9 | | 128 x 4 | 4K byte | " | " |
| SAA6000 | CMOS | 0,04 | | int. | 61 | 122 | | 54 | | 1 | | 2 | | | | 51 | | 3 | 60 | 96 x 4 | 2268 x 8 | für Batteriebetrieb | ITT Intermetall |
| SBPO400A | $I^2L$ | 5 | | ext. | 0,1 | | | 512 | | 1 | | 10 | | | | 4 | | | 40 | | | kaskadierbar | Texas |
| SBPO401A | $I^2L$ | 5 | | ext. | 0,3 | | | 512 | | 1 | | 10 | | | | 4 | | | 40 | | | " | " |
| TMS1000 | PMOS | 0,35 | | int. | | | 15 | 43 | | 1 | 2 | | | | | 23 | | 15 | 28 | 64 x 4 | 1024 x 8 | NMOS und CMOS mit 1,2 MHz, 6 μs, 5 V | Texas |
| TMS1070 | PMOS | 0,35 | | int. | | | 15 | 43 | | 1 | 2 | | | | | 23 | | 15 | 28 | 64 x 4 | 1024 x 8 | wie TMS1000, höhere Ausgangsspannung | " |
| TMS1099 | PMOS | 0,4 | | int. | | | 15 | 43 | | 1 | 2 | | | | | 12 | | 15 | 64 | 64 x 4 | | | " |
| TMS1100 | PMOS | 0,35 | | int. | | | 15 | 43 | | 1 | 2 | | | | | 23 | | 15 | 28 | 128 x 4 | 2048 x 8 | wie TMS1000 | " |
| TMS1200 | PMOS | 0,4 | | int. | | | 15 | 43 | | 1 | 2 | | | | | 25 | | 15 | 40 | 64 x 4 | 1024 x 8 | " | " |
| TMS1270 | PMOS | 0,4 | | int. | | | 15 | 43 | | 1 | 2 | | | | | 25 | | 15 | 40 | 64 x 4 | 1024 x 8 | wie TMS1070 | " |
| TMS1300 | PMOS | 0,4 | | int. | | | 15 | 43 | | 1 | 2 | | | | | 28 | | 15 | 40 | 128 x 4 | 2048 x 8 | wie TMS1000 | " |

## 8bit-μP

| Typ | Technologie | Maschinen-zyklus (MHz) | Takt | | Befehlsausführung (μs) | | | Befehle | Adressierungsarten | Register | | | Stack | Adreßraum (Worte) | DMA | Ein-Ausgänge | Interrupts | Versorgung (V) | Pins | RAM | ROM | Besonderheiten | Hersteller |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | Phasen | ext./int. | kürzeste | längste | mittlere | | | Akkus | Index | allgemein | | | | | | | | | | | |
| 1600 | | | | | | | | | | | | | | | | | | | | | | siehe PIC | |
| 1800 | | | | | | | | | | | | | | | | | | | | | | siehe CDP | |
| 2650 | NMOS | 1,25 | 1 | ext. | 4,8 | 9,6 | | 75 | 8 | 1 | 1 | 7 | 8 x 15 | 32K | X | 8 | 1 | 5 | 40 | | | | Valvo-Signetics, National |
| 2650A | NMOS | 2 | 1 | ext. | 4,8 | 7,2 | | 75 | 8 | 1 | 1 | 7 | 8 x 15 | 32K | X | 8 | 1 | 5 | 40 | | | | " |
| 2650A–I | NMOS | 2 | 1 | ext. | 3 | 9 | | 75 | 8 | 1 | 1 | 7 | 8 x 15 | 32K | X | 8 | 1 | 5 | 40 | | | | " |
| 3850 | NMOS | 2 | 1 | int. | | 13 | 2 | 76 | | 1 | | 64 x 8 | 64 x 8 | 64K | X | 16 | 1 | +5/ +12 | 40 | | | F8-Familie | Fairchild, Mostek, Motorola, SGS/ ATES |
| 3870 | NMOS | 4 | | int. | | 13 | 2 | 76 | | 1 | | RAM | | | | 32 | 2 | 5 | 40 | 64 x 8 | 2K | F8-Ein-Chip-version | Mostek, Fairchild, Motorola |
| 3872 | NMOS | 4 | | int. | | 13 | 2 | 76 | | 1 | | 64 x 8 | | | | 32 | 2 | 5 | 40 | 128 x 8 | 4K | " | " |
| 3873 | NMOS | 4 | | int. | | 13 | 2 | 76 | | 1 | | 64 x 8 | | | | 35 | 4 | 5 | 40 | 64 x 8 | 2K | " | " |
| 3874 | NMOS | 4 | | int. | | 13 | 2 | 76 | | 1 | | 64 x 8 | | | | 32 | 4 | 5 | 40 | 128 x 8 | | " | " |
| 3876 | NMOS | 4 | | int. | | 13 | 2 | 76 | | 1 | | 64 x 8 | | | | 32 | 2 | 5 | 40 | 128 x 8 | 2K | " | " |

| Typ | Technologie | Maschinenzyklus (MHz) | Takt | | Befehlsausführung (μs) | | | Befehle | Adressierungsarten | Register | | | Stack | Adreßraum (Worte) | DMA | Ein-Ausgänge | Interrupts | Versorgung (V) | Pins | RAM | ROM | Besonderheiten | Hersteller |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | Phasen | ext./int. | kürzeste | längste | mittlere | | | Akkus | Index | allgemein | | | | | | | | | | | |
| 6500/1 | NMOS | 2 | 2 | int. | 0,5 | 3,5 | 1,5 | 56 | 13 | 1 | 2 | RAM | 256 | | | 32 | 3 | 5 | 65 | 64 x 8 | 2K | Ein-Chip-μP, nicht erweiterbar | Rockwell, Synertek |
| 6502 | NMOS | 1 | 2 | int. | | | 3 | 56 | 13 | 1 | 2 | | RAM | 64K | | | 3 | 5 | 40 | | | Hauptvertreter der 650X-Familie (vgl. R650X u. R651X mit X = 2 bis 7) | MOS, Rockwell, Synertek |
| 6502A | NMOS | 2 | 2 | int. | | | 1,5 | 56 | 13 | 1 | 2 | | RAM | 64K | | | 3 | 5 | 40 | | | " | " |
| 6516 | NMOS | 2 | | int. | | | 1,5 | 114 | 13 | 1 | 2 | RAM | 256 | | | 8 | 2 | 5 | 40 | | | 16bit-Register, aufwärtskompatibel mit 650X | Synertek |
| 6800 | NMOS | 2 | 2 | ext. | 2 | 5 | | 89 | | 2 | 1 | RAM | | 64K | | 8 | | 5 | 40 | | | | Motorola, AMI, |
| 6802 | NMOS | 4 | 2 | int. | 1 | | | 89 | | 2 | 1 | RAM | | 64K | | 8 | | 5 | 40 | 128 x 8 | | 32Byte RAM nicht-flüchtig | Fairchild, Sescosem, Hitachi, Fujitsu |
| 6801 | NMOS | 2 | 2 | int. | 2 | 5 | | 99 | | 1 | 1 | RAM | | 64K | | 34 | 2 | 5 | 40 | 128 x 8 | 2K | 6800-Ein-Chip-version | Motorola, AMI, Fairchild |
| 6803 | NMOS | 2 | 2 | int. | 2 | 5 | | 99 | | 1 | 1 | RAM | | 64K | | 34 | 2 | 5 | 40 | 128 x 8 | – | " | " |
| 6805 | NMOS | 2 | 2 | int. | 2 | 5 | | 99 | | 1 | 1 | RAM | | – | | 16 | 2 | 5 | 24 28 40 | 64 x 8 | 1,1K | nicht erweiterbar | " |
| 146805 | CMOS | 2 | 2 | int. | | | | 99 | | 1 | 1 | RAM | | – | | | 2 | 5 | 24 | 64 x 8 | 1,1K | CMOS-Version von 6805 | " |
| 6809 | NMOS | 2 | 2 | int. | 2 | 5 | | 100 | | 2 | 2 | RAM | 16 | 64K | | 8 | 1 | 5 | 40 | | | 16bit-Register, aufwärtskompatibel mit 6800 | Motorola, AMI |
| 8008 | PMOS | 0,8 | 2 | ext. | 12,5 | 37,5 | | 48 | | 1 | | 6 | 7 x 14 | 16K | | | 1 | +5/–9 | 18 | | | | Intel, Siemens |
| 8080A | NMOS | 2 | 2 | ext. | 2 | 9 | | 78 | | 1 | | 7 | RAM | 64K | X | 8 | 1 | +5/ +12/ –5 | 40 | | | 16bit-Register | Intel, Siemens, AMD, Texas, NEC, National, |
| 8080A-1 | NMOS | 3,125 | 2 | ext. | 1,3 | 5,75 | | 78 | | 1 | | 7 | RAM | 64K | X | 8 | 1 | " | 40 | | | " | Signetics, |
| 8080A-2 | NMOS | 2,6 | 2 | ext. | 1,5 | 3,75 | | 78 | | 1 | | 7 | RAM | 64K | X | 8 | 1 | " | 40 | | | " | Hitachi |
| 8085 | NMOS | 3 | 2 | int. | 1,3 | 5,85 | | 80 | | 1 | | 7 | RAM | 64K | X | 10 | 5 | 5 | 40 | | | " | Intel, Siemens, AMD, NEC |
| 8085A | NMOS | 5 | 2 | int. | | | 1 | 80 | | 1 | | 7 | RAM | 64K | X | 10 | 5 | 5 | 40 | | | " | " |
| 8021 | NMOS | 3 | | int. | 10 | 20 | | 64 | | 1 | | 2 x 8 | | | | 21 | | 5 | 28 | 64 x 8 | 1K | nicht erweiterbar | Intel |
| 8022 | NMOS | 3 | | int. | 10 | 20 | | 64 | | 1 | | 2 x 8 | | | | 21 | | 5 | 40 | 128 x 8 | 2K | ADC | " |
| 8035 | NMOS | | | | | | | | | | | | | | | | | | | | | | Siemens |
| 8041 | NMOS | 6 | | int. | 2,5 | 5 | | 90 | | 1 | | 2 x 8 | | | | 18 | 1 | 5 | 40 | 64 x 8 | 1K | | Intel |
| 8741 | NMOS | 6 | | int. | 2,5 | 5 | | 90 | | 1 | | 2 x 8 | | | | 18 | 1 | 5 | 40 | 64 x 8 | 1K | EPROM statt ROM | Intel |
| 8048 | NMOS | 6 | | int. | 2,5 | 5 | | 96 | | 1 | | 2 x 8 | | | | 27 | 1 | 5 | 40 | 64 x 8 | 1K | | Intel, Siemens |
| 8748 | NMOS | 6 | | int. | 2,5 | 5 | | 96 | | 1 | | 2 x 8 | | | | 27 | 1 | 5 | 40 | 64 x 8 | 1K | EPROM | " |
| 8049 | NMOS | 6 | | int. | 2,5 | 5 | | 96 | | 1 | | 2 x 8 | | | | 27 | 1 | 5 | 40 | 128 x 8 | 2K | | Intel |
| 8088 | NMOS | 5 | | | 0,4 | | | | 24 | | | 8 | | 1M | | | | 5 | | | | 8086-Architektur | " |
| 8060 | NMOS | 4 | | int. | 5 | 25 | | 46 | 4 | 1 | | | | 4K | X | 10 | | 5 | 40 | | | SC/MP-Familie, 64K adressierbar über Datenbus | National, Signetics |
| 9080A | NMOS | 2 | 2 | int. | 2 | 9 | | 74 | | 1 | | 7 | RAM | 64K | X | 8 | 1 | 5 | 40 | | | 8080-Kompatibel | AMD |
| 9080A-1 | NMOS | 3,125 | 2 | int. | 1,3 | 5,75 | | 74 | | 1 | | 7 | RAM | 64K | X | 8 | 1 | 5 | 40 | | | " | " |
| 9080A-2 | NMOS | 2,6 | 2 | int. | 1,5 | 3,75 | | 74 | | 1 | | 7 | RAM | 64K | X | 8 | 1 | 5 | 40 | | | " | " |
| 9080A-4 | NMOS | 4 | 2 | int. | 1 | 4,5 | | 74 | | 1 | | 7 | RAM | 64K | X | 8 | 1 | 5 | 40 | | | " | " |

| Typ | Technologie | Maschinen-zyklus (MHz) | Takt | | Befehlsausführung (μs) | | | Befehle | Adressierungsarten | Register | | | Stack | Adreßraum (Worte) | DMA | Ein-Ausgänge | Interrupts | Versorgung (V) | Pins | RAM | ROM | Besonderheiten | Hersteller |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | Phasen | ext./int. | kürzeste | längste | mittlere | | | Akkus | Index | allgemein | | | | | | | | | | | |
| 9940E | NMOS | 2,5 | | int. | 2 | 45 | | 68 | | RAM | | RAM | RAM | | | 33 | 3 | 5 | 40 | 128 x 8 | 2K | EPROM | Texas |
| 9940E | NMOS | 5 | | ext. | 1 | 22,5 | | 68 | | RAM | | RAM | RAM | | | 33 | 3 | 5 | 40 | 128 x 8 | 2K | " | " |
| 9940M | NMOS | 2,5 | | int. | 2 | 45 | | 68 | | RAM | | RAM | RAM | | | 33 | 3 | 5 | 40 | 128 x 8 | 2K | " | " |
| 9940M | NMOS | 5 | | ext. | 1 | 22,5 | | 68 | | RAM | | RAM | RAM | | | 33 | 3 | 5 | 40 | 128 x 8 | 2K | " | " |
| 9985 | NMOS | 2 | | int. | 2,5 | | | 68 | | RAM | | RAM | RAM | 64K | | 5 | 3 | 5 | 40 | 256 x 8 | | | " |
| 9985 | NMOS | 4 | | ext. | 1,25 | | | 68 | | RAM | | RAM | RAM | 64K | | 5 | 3 | 5 | 40 | 256 x 8 | | | " |
| 8X300 | STTL | 8 | | int. | | | 0,25 | 8 | | | | 8 | | 8K | | 8 | | 5 | 50 | | | 512 E/A-Einheiten adressierbar | Valvo-Signetics, Fairchild |
| 100K-8bit | | | | | | | | | | | | | | | | | | | | | | | |
| AM9080A | | | | | | | | | | | | | | | | | | | | | | siehe 9080A | AMD |
| CDP 1802CD | CMOS | 3,2 | 1 | int. | | | 5 | 91 | | 1 | | 32 | RAM | 64K | X | | 1 | 5 | 40 | | | 16bit-Register, COSMAC-Familie | RCA, Hughes |
| CDP1802D | CMOS | 6,4 | 1 | int. | | | 2,5 | 91 | | 1 | | 32 | RAM | 64K | X | | 1 | 10 | 40 | | | " | " |
| CDP1803 | CMOS | 6,4 | 1 | int. | | | 2,5 | 91 | | 1 | | 32 | RAM | 64K | X | | 1 | 10 | 28 | | | eingeschränkte Version | " |
| CDP1804 | SOS/ CMOS | 6,4 | 1 | int. | | | 2,5 | 104 | | 1 | | 32 | RAM | 64K | X | | 1 | 10 | 40 | 256 x 8 | 2K | Ein-Chip-Version | " |
| CMP8 | NMOS | | | | | | 2,4 | 50 | | 1 | | 4 | RAM | 64K | | | 2 | +5/ +9/ –5 | 40 | | | | National |
| F8 | | | | | | | | | | | | | | | | | | | | | | siehe 3850 | Fairchild |
| IMP8 | PMOS | | | | | | 4,2 | 38 | | 1 | | 4 | 16 | 64K | X | | 1 | +5 –12 | 24 | | | | National |
| INS8080A | | | | | | | | | | | | | | | | | | | | | | siehe 8080A | National |
| INS8050 | NMOS | 6 | | int. | 2,5 | 5 | | | | | | | | | | | | | | 0,25K | 4K | | " |
| INS8060 | | | | | | | | | | | | | | | | | | | | | | siehe 8060 | " |
| ISP-8A/600 | NMOS | 4 | | int. | 1 | | | 46 | 4 | 1 | | | | 4K | X | 10 | 1 | 5 | 40 | | | SC/MP-II, 64K adressierbar über Datenbus | National |
| LP8000 | PMOS | 0,8 | | ext. | 5 | 30 | | 48 | | | | 48 | | 16K | | 8 | 1 | +5/ –12 | 40 | | | | General Instruments |
| MC6800 | | | | | | | | | | | | | | | | | | | | | | siehe 6800 | Motorola |
| MC6801 | | | | | | | | | | | | | | | | | | | | | | siehe 6801 | " |
| MCS650X | | | | | | | | | | | | | | | | | | | | | | siehe 650X | MOS-Techn. |
| MD46802 | CMOS | 5 | 2 | int. | | | 1 | 89 | | 2 | 1 | RAM | | 64K | | 8 | | 5 | 40 | 128 x 8 | | vgl. 6802 | Motorola |
| μPD8080A | | | | | | | | | | | | | | | | | | | | | | siehe 8080A | NEC |
| μPD8085A | | | | | | | | | | | | | | | | | | | | | | siehe 8085A | " |
| μPD8035 | | | | | | | | | | | | | | | | | | | | | | siehe 8035 | " |
| μPD8041 | | | | | | | | | | | | | | | | | | | | | | siehe 8041 | " |
| μPD8048 | | | | | | | | | | | | | | | | | | | | | | siehe 8048 | " |
| μPD780 | | | | | | | | | | | | | | | | | | | | | | siehe Z80 | " |

8bit-$\mu$P

| Typ | Technologie | Maschinen-zyklus (MHz) | Takt | | Befehlsausführung ($\mu$s) | | | Befehle | Adressierungsarten | Register | | | Stack | Adreßraum (Worte) | DMA | Ein-Ausgänge | Interrupts | Versorgung (V) | Pins | RAM | ROM | Besonderheiten | Hersteller |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | Phasen | ext./int. | kürzeste | längste | mittlere | | | Akkus | Index | allgemein | | | | | | | | | | | |
| PIC1645 | NMOS | 1 | 2 | int. | | | 4 | | | | | 32 | 3 | | | 12 | 1 | 5 | 18 | 28 x 8 | 256 x 12 | Einchip-$\mu$P, nicht erweiterbar | General Instr., EMM |
| PIC1650 | NMOS | 1 | 2 | int. | | | 4 | | | | | 32 | 3 | | | 32 | | 5 | 40 | 32 x 8 | 512 x 12 | " | " |
| PIC1655 | NMOS | 1 | 2 | int. | | | 4 | | | | | 32 | 3 | | | 20 | | 5 | 40 | 32 x 8 | 512 x 12 | " | " |
| PIC1664 | NMOS | 1 | 2 | int. | | | 4 | | | | | 32 | 3 | | | 32 | | 5 | 64 | 32 x 8 | | Einchip-$\mu$P, ohne ROM | General Instr., |
| PIC1670 | NMOS | 1 | 2 | int. | | | 4 | | | | | 32 | 4 | | | 32 | | 5 | 40 | 48 x 8 | 1K | nicht erweiterbar | " |
| PPS-8 | PMOS | 0,25 | 4 | ext. | 5 | 15 | 4 | 109 | | 1 | | 2 | | 32K | X | | 3 | +5/ −12 | 42 | | | | Rockwell, Telefunken |
| PPS-8/2 | PMOS | 0,25 | 4 | ext. | 5 | 15 | 4 | 109 | | 1 | | 2 | | 32K | X | 18 | 3 | +5/ −12 | 52 | 64 x 8 | 2K | Zweichip-$\mu$P | " |
| R6500 | NMOS | 2 | 2 | int. | 1 | 3 | | 56 | 13 | 1 | 2 | | RAM | | | 27 | 3 | 5 | 40 | 128 x 8 | 2K | | Rockwell |
| R6500-1 | NMOS | 2 | 2 | int. | 1 | 3 | | 56 | 13 | 1 | 2 | | RAM | | | 34 | 3 | 5 | 40 | 64 x 8 | 2K | | " |
| R6502 | NMOS | 2 | 2 | int. | 1 | 3 | | 56 | 13 | 1 | 2 | | RAM | 64K | | | 3 | 5 | 40 | | | | " |
| R6512 | NMOS | 2 | 2 | ext. | | | | 56 | 13 | 1 | 2 | | RAM | 64K | | | 3 | 5 | 40 | | | | " |
| R6503 | NMOS | 2 | 2 | int. | 1 | 3 | | 56 | 13 | 1 | 2 | | RAM | 4K | | | 3 | 5 | 28 | | | | " |
| R6513 | NMOS | 2 | 2 | ext. | | | | 56 | 13 | 1 | 2 | | RAM | 4K | | | 3 | 5 | 28 | | | | " |
| R6504 | NMOS | 2 | 2 | int. | 1 | 3 | | 56 | 13 | 1 | 2 | | RAM | 8K | | | 1 | 5 | 28 | | | | " |
| R6514 | NMOS | 2 | 2 | ext. | | | | 56 | 13 | 1 | 2 | | RAM | 8K | | | 1 | 5 | 28 | | | | " |
| R6505 | NMOS | 2 | 2 | int. | 1 | 3 | | 56 | 13 | 1 | 2 | | RAM | 4K | | | 1 | 5 | 28 | | | | " |
| R6515 | NMOS | 2 | 2 | ext. | | | | 56 | 13 | 1 | 2 | | RAM | 4K | | | 1 | 5 | 28 | | | | " |
| R6506 | NMOS | 2 | 2 | int. | 1 | 3 | | 56 | 13 | 1 | 2 | | RAM | 4K | | | 1 | 5 | 28 | | | | " |
| R6507 | NMOS | 2 | 2 | int. | 1 | 3 | | 56 | 13 | 1 | 2 | | RAM | 8K | | | | 5 | 28 | | | | " |
| SAB8080 | | | | | | | | | | | | | | | | | | | | | | siehe 8080A | Siemens |
| SAB8085 | | | | | | | | | | | | | | | | | | | | | | siehe 8085A | " |
| SAB8048 | | | | | | | | | | | | | | | | | | | | | | siehe 8048 | " |
| S6800 | | | | | | | | | | | | | | | | | | | | | | siehe 6800 | AMI |
| SC/MP | | | | | | | | | | | | | | | | | | | | | | siehe 8060 und ISP-8A/600 | National |
| SMS300 | TTL | 10 | 1 | | | | | 8 | | | | | | 8K | | | | | 50 | | | | Scientific |
| TMS8080A | | | | | | | | | | | | | | | | | | | | | | siehe 8080A | Texas |
| Z8 | NMOS | 4 | | int. | 1,5 | 2,25 | | 129 | 7 | | | 124 | X | 2X 64K | X | 32 | 8 | 5 | 40 | 144 x 8 | 2K | Einchip-$\mu$P | Zilog |
| Z80 | NMOS | 2,5 | 1 | ext. | 1,6 | 9,2 | 3 | 158 | | | | 14 | RAM | 64K | X | | 1 | 5 | 40 | | | enthält 8080-Befehlssatz | Zilog |
| Z80A | NMOS | 4 | 1 | ext. | 1 | 5,75 | 2 | 158 | | | | 14 | RAM | 64K | X | | 1 | 5 | 40 | | | " | " |

12bit-$\mu$P

| Typ | Technologie | Maschinen-zyklus (MHz) | Takt | | Befehlsausführung ($\mu$s) | | | Befehle | Adressierungsarten | Register | | | Stack | Adreßraum (Worte) | DMA | Ein-Ausgänge | Interrupts | Versorgung (V) | Pins | RAM | ROM | Besonderheiten | Hersteller |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | Phasen | ext./int. | kürzeste | längste | mittlere | | | Akkus | Index | allgemein | | | | | | | | | | | |
| 6100 | | | | | | | | | | | | | | | | | | | | | | siehe IM6100 | |
| IM6100 | CMOS | 4 | 1 | int. | 2,5 | 5,5 | | 81 | | 1 | | 0 | RAM | 4K | X | | 1 | 5 | 40 | | | PDP-8-Befehlssatz | Intersil, Harris |
| IM6100A | CMOS | 8 | 1 | int. | 1,25 | 2,75 | | 81 | | 1 | | 0 | RAM | 4K | X | | 1 | 10 | 40 | | | " | " |
| IM6100C | CMOS | 3,3 | 1 | int. | 3 | 6,7 | | 81 | | 1 | | 0 | RAM | 4K | X | | 1 | 5 | 40 | | | " | " |
| T3190 | NMOS | 2,5 | 1 | int. | 10 | 30 | | 108 | | 1 | | 8 | RAM | 4K | X | | 8 | +5/–5 | 36 | | | | Toshiba |
| TLCS12 | PMOS | | | | 10 | | | 80 | | 1 | | 6 | RAM | 4K | | | 8 | +5/–5 | 36 | | | | Toshiba/Harris |

| Typ | Technologie | Maschinen-zyklus (MHz) | Takt | | Befehlsausführung (μs) | | | Befehle | Adressierungsarten | Register | | | Stack | Adreßraum (Worte) | DMA | Ein-Ausgänge | Interrupts | Versorgung (V) | Pins | RAM | ROM | Besonderheiten | Hersteller |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | Phasen | ext./int. | kürzeste | längste | mittlere | | | Akkus | Index | allgemein | | | | | | | | | | | |
| 68000 | NMOS | | 1 | ext. | | | 1 | 61 | 10 | | | 16 x 32 | | 16M | X | | | 5 | 64 | | | 32bit-ALU und Register | Motorola |
| 8086 | NMOS | 5 | | ext. | 0,6 | 3,4 | 0,8 | | 5 | | | 8 | | 1M | | | | 5 | 40 | | | | Intel |
| 8086 | NMOS | 8 | | ext. | 0,37 | 2,1 | 0,5 | | 5 | | | 8 | | 1M | | | | 5 | 40 | | | | " |
| 8900 | NMOS | 2 | 2 | ext. | 2,5 | 5 | | 45 | | | | 4 | 10 | 64K | X | | 6 | +5/ +12/ −5 | 40 | | | PACE-Familie | National |
| 9440/8 | $I^2L$ | 8 | 1 | int. | | | 1,25 | 183 | 8 | | | 8 | RAM | 32K | X | 64 | X | 5 | 40 | | | Microflame-Familie, 4bit-ALU | Fairchild |
| 9440/10 | $I^2L$ | 10 | 1 | int. | | | | 183 | 8 | | | 8 | RAM | 32K | X | 64 | X | 5 | 40 | | | " | " |
| 9440/12 | $I^2L$ | 12 | 1 | int. | | | | 183 | 8 | | | 8 | RAM | 32K | X | 64 | X | 5 | 40 | | | " | " |
| 9445 | $I^2L$ | 10 | 1 | int. | | | | 183 | | | | | | | | | | | | | | 16bit-ALU | " |
| 9900 | | | | | | | | | | | | | | | | | | | | | | siehe SBP9900 und TMS9900 | Texas |
| CP-1600A | NMOS | 4 | 2 | X | 1,6 | 4,8 | | 87 | | | | 8 | RAM | 64K | X | | 1 | +12/ +5/−3 | 40 | | | 8bit-ALU, PDP11-Befehle | General Instr., EMM |
| CP-1610 | NMOS | 2 | 2 | X | 4 | 12 | | 87 | | | | 8 | RAM | 64K | X | | 1 | " | 40 | | | " | " |
| F100L | TTL | 20 | | | 1,2 | 5,75 | | 28 | | | | | RAM | 32K | X | | 1 | 5 | 40 | | | | Ferranti |
| IMP-16 | NMOS | 0,7 | | | 4,9 | 10 | | 42 | 4 | | | 4 | 16 | 64K | | | | +5/ −12/ −10 | 24 | | | | National |
| INS8900 | | | | | | | | | | | | | | | | | | | | | | siehe 8900 | National |
| IPC-16A | PMOS | | | ext. | | | 5 | 45 | 4 | | | 4 | 10 | | | | 6 | +5/ −12 | 40 | | | Einchip-μP, PACE-Familie | National |
| Micro Nova | | | | | | | | | | | | | | | | | | | | | | siehe mN601 | Data General |
| Miproc 16 | STTL | 2,85 | | int. | 0,35 | 11,2 | | 82 | | | 1 | 2 | | 128 K | X | 32 | 8 | 5 | | | | | Plessey |
| Miproc 16F | STTL | 4 | | int. | 0,25 | | | 82 | | | 1 | 2 | | 128 K | X | 32 | 8 | 5 | | | | | " |
| mN601 | NMOS | 8,33 | 2 | int. | 1,2 | 29,5 | | 42 | | | | 4 | RAM | 32K | X | 47 | 1 | +5/ +10/ +14/ −4,25 | 40 | | | Nova 3-Befehls-satz | Data General |
| MN1610 | NMOS | 2 | 2 | ext. | 2 | 6 | | 33 | | | | 5 | RAM | 64K | X | | 3 | +5/ +12/ −3 | 40 | | | | Panafacom |
| PACE | | | | | | | | | | | | | | | | | | | | | | siehe IPC-16A | National |
| SAB8086 | | | | | | | | | | | | | | | | | | | | | | siehe 8086 | Siemens |
| SBP9900 | $I^2L$ | 3 | 1 | ext. | 2 | 31 | | 69 | | | | 16 | RAM | 64K | X | | 16 | 5 | 64 | | | | Texas |
| TMS9900 | NMOS | 3 | 4 | ext. | 4,6 | 10,6 | | 69 | | | | 16 | RAM | 64K | X | | 16 | +12/ +5/−5 | 64 | | | | " |
| TMS9940 | NMOS | 5 | | int. | 2,4 | 45 | | 58 | | | | 16 | | | | 32 | 4 | 5 | 40 | 128 x 8 | 2K | Einchipversion, nicht erweiterbar, EPROM | " |
| TMS9980A | NMOS | 10 | 1 | ext. | 8,8 | 20,8 | | 69 | | | | 16 | RAM | 16K | X | | 4 | +12/ +5/−5 | 40 | | | | " |
| TMS9981 | NMOS | 10 | | int. | 8,8 | 20,8 | | 69 | | | | 16 | RAM | 16K | X | | 4 | +12/ +5 | 40 | | | | " |

| Typ | Technologie | Maschinenzyklus (MHz) | Takt | | Befehlsausführung (μs) | | | Befehle | Adressierungsarten | Register | | | Stack | Adreßraum (Worte) | DMA | Ein-Ausgänge | Interrupts | Versorgung (V) | Pins | RAM | ROM | Besonderheiten | Hersteller |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | Phasen | ext./int. | kürzeste | längste | mittlere | | | Akkus | Index | allgemein | | | | | | | | | | | |
| WD-16 | NMOS | 3,3 | 4 | ext. | 2,1 | | | 116 | | | | 6 | RAM | 64K | X | | 16 | +12/ +5/–5 | 40 | | | | Western Digital |
| Z8000 | NMOS | 4 | 1 | ext. | 0,75 | 17,5 | 3 | 110 | 8 | | | 16 | RAM | 64K | X | | | 5 | 40 | | | 32bit-Operationen | Zilog |
| Z8000 | NMOS | 4 | 1 | ext. | 0,75 | 17,5 | 3 | 110 | 8 | | | 16 | RAM | 8M | X | | | 5 | 64 | | | " | " |


ENGLISH FOR INTERNATIONAL CONFERENCES ist ein mehrmedialer Sprachkurs, der mit Unterstützung des „Ernst Hellmut Vits-Fonds zur Förderung des Führungsnachwuchses in Stiftungen und Wissenschaftsorganisationen" speziell für den Bedarf von Fachleuten aus Wissenschaft, Wirtschaft, Politik und Verwaltung entwickelt wurde. Der „Ernst Hellmut Vits-Fonds" wird vom Stifterverband für die Deutsche Wissenschaft treuhänderisch verwaltet.
Verfasser des Sprachkurses ist der Fachautor für Englisch als Fremdsprache Anthony Fitzpatrick.
Ziel des Sprachkurses ENGLISH FOR INTERNATIONAL CONFERENCES ist es, dem angesprochenen Adressatenkreis die ***sprachlich-aktive*** Teilnahme an internationalen Konferenzen – Tagungen, Kongressen, Symposien aller Art mit Plenar- und Ausschußsitzungen sowie Arbeitsgruppen – zu ermöglichen.
Voruntersuchungen mit einer eigens dafür entwickelten Piloteinheit ergaben, daß es nicht so sehr die Fachterminologie des eigenen Arbeitsbereichs ist, die Schwierigkeiten bereitet, sondern vielmehr der flüssige Gebrauch des Alltagsenglisch in Verbindung mit jenen speziellen ***Redestrategien***, die zur „Bewältigung" internationaler Konferenzsituationen notwendig sind.
Der Schwerpunkt des Sprachkurses liegt dementsprechend auf der ***gesprochenen Sprache***. Ausgehend von vorhandenen Grundkenntnissen in Wortschatz und Grammatik trainiert der Sprachkurs beim Lerner die Fähigkeit, einer Konferenzsituation bei normalem muttersprachlichen Sprechtempo zu folgen und adäquat in der englischen Sprache zu reagieren.
Der Sprachkurs ENGLISH FOR INTERNATIONAL CONFERENCES besteht aus vier in sich abgeschlossenen Einheiten, die in vier Tonkassetten mit je 60 Minuten Spieldauer angeboten werden;

- **Reise und Ankunft (Unit 1: Travelling)**
- **Gespräche am Rande (Unit 2: Making Social Contacts)**
- **Konferenz A: Plenarsitzung (Unit 3: Conference A, Plenary Session)**
- **Konferenz B: Arbeitsgruppen (Unit 4: Conference B, Working Groups).**

Jede Einheit besteht aus einem der Situation entsprechenden lebensnahen Hörspiel, das der Lernende nach ein- oder mehrmaligem Abhören geleitet von einem Moderator in mehreren Übungen aktiv, d. h. ***selbstsprechend*** durcharbeitet. Dabei werden in Dialogform die erforderlichen Sprechintentionen und Redestrategien genauso geübt wie das aktive Zuhören. Einige Besonderheiten der englischen Grammatik, die erfahrungsgemäß selbst Könnern Schwierigkeiten bereiten, werden besonders intensiv trainiert. Für die Tonaufnahmen des Sprachkurses wurden ***Berufssprecher mit englischer Muttersprache*** eingesetzt.
Die Tonkassetten werden von einer Broschüre begleitet, die alle Texte enthält und der raschen Wiederholung dient; sie sollte immer dann herangezogen werden, wenn der Lernende einen bestimmten Abschnitt besonders intensiv üben will. In der Begleitbroschüre endet jede Einheit mit einem Selbsttest, der zeigen soll, ob ein nochmaliges Durcharbeiten der Einheit erforderlich ist.
Der Aufbau des Sprachkurses ist so angelegt, daß die Tonkassetten auch ohne die Begleitbroschüre durchgearbeitet werden können.
Die Erläuterungen sowohl auf den Tonkassetten als auch in der Begleitbroschüre sind ausschließlich in Englisch abgefaßt.

Anthony Fitzpatrick
**ENGLISH FOR INTERNATIONAL CONFERENCES**
A LANGUAGE COURSE FOR THOSE WORKING IN THE FIELDS OF SCIENCE, ECONOMICS, POLITICS AND ADMINISTRATION. 1978. 4 C-60 Cassetten und Begleitbuch im Schuber. 84,– DM (fPr)

**Lassen Sie sich den Kurs ganz unverbindlich von Ihrem Buchhändler vorführen oder fordern Sie ausführliches Informationsmaterial mittels der am Schluß eingehefteten Bestellkarte an.**

## 1.4 Mikrocomputer

Ebenso wie Taschenrechner, aber anders als Mikroprozessoren, sind Mikrocomputer in der Regel unter ihrem Herstellernamen bekannt. Auch solche Beispiele wie „PET 2001" beweisen nicht das Gegenteil; denn nahezu jedem fällt dabei der Hersteller Commodore ein. Und so sind die folgenden Datentabellen alphabetisch nach Herstellern geordnet.

In der Auflistung ist danach der Mikroprozessor genannt, bei Entwicklungssystemen auch mehrere, wenn das System mehrere unterstützt. Eine besondere Bedeutung kommt der Spalte „Art" zu. Hier ist angegeben, für welchen Hauptzweck der µC ausgelegt ist. Die dafür verwendeten Abkürzungen haben folgende Bedeutung:

BLC – *Board Level Computer*; Computer, dessen einzelne Funktionseinheiten auf getrennten Platinen aufgebaut sind. Damit wird eine individuelle Zusammenstellung der benötigten Konfiguration möglich (modularer Aufbau). Manchmal wird auch MBC (*Multi Board Computer*) benutzt.

SBC – *Single Board Computer*; funktionsfähiger µC auf einer Platine (CPU, Ein-Ausgabe und ein Minimum an Speicher).

EBC – *Euro-Board Computer*; SBC im Europakarten-Format (100 mm × 160 mm). Aber auch ein BLC kann mit Europakarten aufgebaut sein. Dann ist dies in der Spalte „Besonderheiten" vermerkt.

KIT – Mikrocomputer-Bausatz.

LS – Lernsystem; d. h. in der Regel, µC in Minimalausstattung mit Monitor, etwas RAM, LED-Anzeige und Hexadezimaltastatur.

ES – Entwicklungssystem, also sehr weit ausgebautes und komfortables µC-System, das eine bequeme Programmentwicklung erlaubt.

GPC – *General Purpose Computer*; Mehrzweckcomputer. Damit bezeichnet man µC, die so vollständig sind und beispielsweise in BASIC programmiert werden können, daß sie in sehr verschiedenen Anwendungsbereichen sofort einsetzbar sind. Hierzu gehören auch alle die Systeme, die oft *Personal Computer, Home Computer* oder *Hobby Computer* genant werden.

In den nachfolgenden Spalten kommen eine Reihe weiterer Abkürzungen vor:

HEX – Hexadezimal-Tastatur

OCT – Oktal-Tastatur

BIN – Binärschalter

ASCII – Schreibmaschinen-Tastatur zum Eingeben von ASCII-Zeichen

INT – Interface für die betreffende Spezifikation vorhanden

ASS – In Assembler programmierbar

FORTR – FORTRAN

SPEZI – Spezialbus des Systems

MULTI –
SYS65 –
POLY –
EXOR –
Q –
HP –
ESCO –
MCS –
} Spezielle Hersteller-Busse

S50 – Standardbus 50

S100 – Standardbus 100

PROM PROGR – Einrichtung zum Einbrennen von Programmen in EPROM (Programmer)

DMA – *Direct Memory Access*, direkter Speicherzugriff

ADC – Analog-Digital-Converter

| Hersteller (Distributor) Typ | µP | Art | Tastatur | Anzeige (Stellen) | Bildschirm | Drucker | Netzteil | Monitor (byte) | Sprachen | Speicher -ausbau (byte) RAM | ROM | PROM | direkt | mit Zusatz | Kassettenrecorder Anschlüsse | eingebaut | Schnittstellen V 24 | 20 mA | TTL | IEC | Systembus | Besonderheiten |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **ACORN** (BETZ) | | | | | | | | | | | | | | | | | | | | | | |
| ACORN | 6502 | KIT | HEX | 8 | | | | 0,75K | | 1K | 0,5K | | 4K | | 1 | | | X | | | | |
| **ADCOMP** | | | | | | | | | | | | | | | | | | | | | | |
| TEMP8 | 3870<br>3872<br>6800<br>F-8<br>Z-80 | ES | HEX | 6 | | | X | | | | | | | | | | | | | | | PROM-PROGR |
| **AMC** (Siemens) | | | | | | | | | | | | | | | | | | | | | | |
| AMC95/4000 | 9080 | SBC | | | | | | | | 4K | | | 12K | 64K | | | X | X | | | MULTI | DMA, Arithmetik |
| ECB85 | 8085 | LS | HEX | 8 | | | | 2K | | 1,25K | | | 2K | | X | | | | | | | EPROM-PROGR |
| SKC85 | 8085 | EBC | | | | | | | | 0,75K | | | 4K | 32K | | | X | X | | | SPEZI | 0,25K RAM permanent |
| SMP80 | 8080 | BLC | | | | | | 1K | | | | | | 64K | | | X | X | | | SPEZI | DMA, EBC |
| **AMD** (Elbatex) | | | | | | | | | | | | | | | | | | | | | | |
| AM2900K1 | 2900 | LS | | | | | | | | | | | | | | | | | | | | |
| **AMI** (Neumüller) | | | | | | | | | | | | | | | | | | | | | | |
| AMI6800 | 6800 | SBC | | | | | | | | 1K | 2K | 2K | 4K | | | | X | X | | | | DMA, PROM-PROGR |
| EVK300 | 6800 | SBC | | | INT | INT | | | ASS, BASIC | 1K | 6K | | 14K | 64K | | | X | X | | | SPEZI | DMA, PROM-PROGR |
| **Analog Precision** | | | | | | | | | | | | | | | | | | | | | | |
| TPM200 | 8080 | SBC | | | | | | | ASS | 0,25K | 1,2K | | 16K | | | | | | | | SPEZI | 8 bit ADC |
| **Apple** (BASIS) | | | | | | | | | | | | | | | | | | | | | | |
| A2B0004X | 6502 | SBC | | | | | | | ASS, BASIC | 4K | | | 64K | | 1 | | | | X | | SPEZI | |
| Apple I | 6502 | GPC | ASCII | | INT | | X | | | 4K | | | | | | X | | | | | SPEZI | |
| Apple II | 6502 | GPC | ASCII | | INT | INT | X | 2K | ASS, BASIC | 4K | 8K | | 64K | | | X | X | X | | | SPEZI | DMA |
| **ATARI** | | | | | | | | | | | | | | | | | | | | | | |
| ATARI400 | 6502 | GPC | ASCII | | INT | | X | | BASIC | 8K | 8K | | 24K | | | | X | | | | | Grafik |
| ATARI800 | 6502 | GPC | ASCII | | INT | | X | | BASIC | 8K | 8K | | 48K | | | 2 | X | | | | | Grafik, Floppy-INT |
| **Applied Systems** | | | | | | | | | | | | | | | | | | | | | | |
| ASC/80 | 6800<br>8085<br>Z80 | SBC | INT | | INT | | X | | ASS, BASIC, FORTR, PL/M | 0,5K | 4K | | 6,5K | 64K | | X | | X | | | S-100 | |
| **Bally** | | | | | | | | | | | | | | | | | | | | | | |
| Arcade | Z80 | GPC | ASCII | | INT | | | | BASIC | 4K | 8K | | 20K | 44K | | | | | | X | | TV-Spiele |
| **Bedford** | | | | | | | | | | | | | | | | | | | | | | |
| MCS | 8080 | SBC | INT | | | INT | | | | 1K | 7K | | | 64K | | | X | X | | | | |
| **Canon** | | | | | | | | | | | | | | | | | | | | | | |
| BX-1 | | GPC | ASCII | 16 | | X | X | | BASIC | | | | 16K | | | | X | | | | | Floppy |

| Hersteller (Distributor) Typ | µP | Art | Tastatur | Anzeige (Stellen) | Bildschirm | Drucker | Netzteil | Monitor (byte) | Sprachen | Speicher RAM | Speicher ROM | Speicher PROM | -ausbau (byte) direkt | -ausbau (byte) mit Zusatz | Kassetten-recorder Anschlüsse | Kassetten-recorder eingebaut | Schnitt-stellen V 24 | Schnitt-stellen 20 mA | Schnitt-stellen TTL | Schnitt-stellen IEC | Systembus | Besonderheiten |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Central Data** | | | | | | | | | | | | | | | | | | | | | | |
| 2650 System | 2650 | SBC | INT | | INT | | | | ASS, BASIC | 2K | 1K | | 6K | | | X | | | | | S-100 | |
| **Christiani** | | | | | | | | | | | | | | | | | | | | | | |
| µP-Labor | SC/MP | LS | HEX | 6 | | | X | X | | 0,5K | 1K | | 2K | | 1 | | | | | | | Fernlehrgang |
| **Commodore** | | | | | | | | | | | | | | | | | | | | | | |
| 2001 | 6502 | GPC | ASCII | | X | | X | 6K | BASIC | 8K | 14K | | 32K | | | X | | | | X | | |
| CBM3001 | 6502 | GPC | ASCII | | X | INT | X | 6K | BASIC FORTR. | 32K | 14K | | | | | | | | | X | | Floppy-INT |
| **COMPAS** | | | | | | | | | | | | | | | | | | | | | | |
| CSB2 | 6502 | SBC | | | | | | | ASS, CSL65 | 2K | 8K | | 44K | 64K | | | X | X | | | SYS65 | DMA |
| **Compucolor** | | | | | | | | | | | | | | | | | | | | | | |
| Comp. II | 8080 | GPC | ASCII | | X | | X | | | 8K | 16K | | 64K | | X | | X | | | | | Mini-Floppy, Color-Graphik |
| **Compucorp** | | | | | | | | | | | | | | | | | | | | | | |
| 625 | Z80 | GPC | ASCII | | X | X | X | | ASS, BASIC | 56K | | | | | | | | | | X | | Doppel-Floppy |
| **Computer Automation** (Geveke) | | | | | | | | | | | | | | | | | | | | | | |
| LSI4/10 | 16bit | SBC | INT | | INT | INT | | | ASS, BASIC, FORTR. | 4K | 4K | | | 64K | | | X | X | | | SPEZI | DMA, NAKED-MINI-Familie |
| **Control Logic** | | | | | | | | | | | | | | | | | | | | | | |
| CCS-1025 | 8080 | SBC | INT | | INT | INT | | | ASS, FORTR. | 1K | | | 6K | | | | | X | | | | |
| CCS-1143 | Z80 | SBC | INT | | INT | INT | | | ASS, FORTR. | 1K | | | 16K | 64K | | | X | X | | | POLY | |
| MMI-MSC | Z80 | SBC | | | | | | | ASS, FORTR. | 1K | | | 2K | 64K | | | X | X | | | POLY | |
| **Control Systems** | | | | | | | | | | | | | | | | | | | | | | |
| UDS470 | 6800 | GPC | INT | | INT | | X | X | ASS, PASCAL | 32K | | 16K | | | | | | | | | | Doppel-Floppy |
| **Cramer** | | | | | | | | | | | | | | | | | | | | | | |
| 8080 Cramerkit | 8080 | KIT | INT | 6 | | | | X | | 1K | | | | | 1 | | X | X | | | S-100 | |
| 8080A Cramerkit | 8080A | KIT | INT | 6 | | | | X | | 1K | | | | | | X | X | X | | | S-100 | |
| 6800 Cramerkit | 6800 | KIT | HEX | 6 | | | | X | | 1K | | | | | | X | X | X | | | S-100 | |
| F8 Cramerkit | F8 | KIT | HEX | 6 | | | | X | | 1K | | | | | | X | X | X | | | S-100 | |
| CosmacCramerkit | 1802 | KIT | HEX | 6 | | | | X | | 1K | | | | | | X | X | X | | | S-100 | |
| 16bit µC KIT | 9900 | KIT | INT | | INT | | | X | | 16K | | | | | 1 | | X | X | | | S-100 | Floppy-INT |
| **Creative Micro Systems** | | | | | | | | | | | | | | | | | | | | | | |
| 9600 | 6802 | SBC | | | | | | | ASS | 1,1K | 2K | | 10K | 64K | | | X | | | | EXOR | DMA |

| Hersteller (Distributor) Typ | µP | Art | Tastatur | Anzeige (Stellen) | Bildschirm | Drucker | Netzteil | Monitor (byte) | Sprachen | Speicher RAM | Speicher ROM | Speicher PROM | -ausbau (byte) direkt | -ausbau (byte) mit Zusatz | Kassettenrecorder Anschlüsse | Kassettenrecorder eingebaut | Schnittstellen V 24 | Schnittstellen 20 mA | Schnittstellen TTL | Schnittstellen IEC | Systembus | Besonderheiten |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Cromemco** (Digitronic) | | | | | | | | | | | | | | | | | | | | | | |
| CS-3 | Z80 | ES | | | INT | INT | X | | ASS, BASIC, FORTR, COBOL | 32K | | | | | | | | | | | | Doppel-Floppy, PROM-PROGR |
| SCC-W | Z80 | SBC | INT | | INT | | | | BASIC | 1K | | | 10K | 64K | | | X | X | | | S-100 | DMA, Floppy-INT |
| Z-1 | Z80 | GPC | INT | | | | X | | | 8K | | | | | | | | | | | S-100 | |
| Z-2 | Z80 | BLC | INT | | INT | | X | | BASIC, FORTR. | 32K | | 1K | | 64K | | | X | X | | | S-100 | Doppel-Floppy |
| **Data General** | | | | | | | | | | | | | | | | | | | | | | |
| 8563 | mN601 | SBC | INT | | INT | INT | | | ASS, BASIC | 4K | | | 8K | 32K | | | X | X | | | SPEZI | 16bit-µC, Floppy- und Platten-INT |
| MBC/1 | mN601 | SBC | INT | | INT | INT | | | ASS, BASIC, FORTR. | 1K | | | 4K | 32K | | | X | X | | | SPEZI | " |
| Micro Nova | mN601 | SBC | INT | | INT | | | | ASS, BASIC | 2K | 0,5K | | 10K | 32K | | X | | | | | S-100 | " |
| **DATARAM** | | | | | | | | | | | | | | | | | | | | | | |
| LSI-11KIT | LSI-11 | KIT | | | | | | | | | | | | | | | | | | | | vgl. Digital Equipment |
| **Dietz** | | | | | | | | | | | | | | | | | | | | | | |
| Micro621 | 621 | BLC | INT | | INT | INT | X | | HLL | 4K | | | | 52K | | | X | X | | | INT | bis 20 Peripheriegeräte |
| **Digital Electronics** | | | | | | | | | | | | | | | | | | | | | | |
| DE-68 | 6800 | GPC | HEX | | | | | X | | 1K | | | | | | | | | | | S-100 | |
| DE-68DT | 6800 | GPC | ASCII | 6 | INT | X | | 6K | ASS, BASIC, FORTR. | 64K | | | | | 1 | | X | | | | S-100 | Floppy |
| **Digital Equipment** | | | | | | | | | | | | | | | | | | | | | | |
| LSI-11 | 1600 | BLC | INT | | INT | INT | | | | 4K | | | | | 1 | | | | | | Q | |
| LSI-11/23 | LSI-11 | SBC | INT | | INT | INT | | | BASIC, FORTR. | 64K | | | | 256K | | | X | X | | | Q | |
| KD11-F | LSI-11 | SBC | INT | | INT | INT | | | ASS, BASIC, FORTR., APL | 4K | | | 8K | 32K | | X | X | X | | | Q | Floppy- und Platten-INT, DMA |
| KD11-HB | LSI-11/2 | BLC | INT | | INT | INT | | | " | 8K | | | 16K | 32K | | X | X | X | | | Q | " |
| KD11-HU | LSI-11/2 | BLC | INT | | INT | INT | | | " | 0,25K | 4K | | 8,5K | 32K | | X | X | X | | | Q | " |
| KD11-J | LSI-11 | BLC | INT | | INT | INT | | | " | 4K | | | 8K | 32K | | X | X | X | | | Q | " |
| **Digital Group** (Micro Mail) | | | | | | | | | | | | | | | | | | | | | | |
| Bytemaster 1 | Z80 | GPC | ASCII | | X | INT | X | | | 18K | | | | 64K | 1 | | X | X | | | | |
| Bytemaster 3 | Z80 | GPC | ASCII | | X | INT | X | | | 18K | | | | 64K | | | X | X | | | | Doppel-Floppy |
| **Digitronic** | | | | | | | | | | | | | | | | | | | | | | |
| ELDIS80 | Z80 | GPC | ASCII | | X | INT | X | 1K | BASIC | 32K | | | | 512K | | | | X | | | | Floppy-, Platten-INT |
| **EDS** | | | | | | | | | | | | | | | | | | | | | | |
| MACH I | Z-80 | GPC | ASCII | | X | INT | X | | BASIC | 16K | 8K | | | | | | | | | | | Floppy-INT |
| MACH III | Z-80 | GPC | ASCII | | X | INT | X | | BASIC | 64K | 2K | | | | | | | | | | | Doppel-Floppy |
| **EKF** | | | | | | | | | | | | | | | | | | | | | | |
| System 6800 | 6800 | EBC | | | | | | X | | 0,25K | | 8K | | | | | X | X | | | | |

| Hersteller (Distributor) Typ | µP | Art | Tastatur | Anzeige (Stellen) | Bildschirm | Drucker | Netzteil | Monitor (byte) | Sprachen | Speicher -ausbau (byte) RAM | ROM | PROM | direkt | mit Zusatz | Kassettenrecorder Anschlüsse | eingebaut | Schnittstellen V 24 | 20 mA | TTL | IEC | Systembus | Besonderheiten |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **E & L** | | | | | | | | | | | | | | | | | | | | | | |
| MMD-1 | 8080<br>Z80 | SBC | INT | 6 | INT | | | X | BASIC | 0,5K | 0,5K | | | 64K | | X | | X | | | S-100 | |
| **Elrad** | | | | | | | | | | | | | | | | | | | | | | |
| Triton | 8080 | KIT | ASCII | | INT | | X | 1K | BASIC | 4K | 1K | | | 64K | 1 | | | | | | | |
| **ELTEC** | | | | | | | | | | | | | | | | | | | | | | |
| EUROCOM-1 | 6802 | SBC | HEX | 8 | INT | | | 2K | | 1K | 2K | | | | 1 | | X | | | | | |
| **ESCO** | | | | | | | | | | | | | | | | | | | | | | |
| (Neumüller)<br>ESCO | 6502 | EBC | INT | 6 | INT | INT | | 2K | ASS | 2K | 8K | | | 64K | 1 | | | X | | | ESCO | DMA |
| **Exidy** | | | | | | | | | | | | | | | | | | | | | | |
| (Computershop)<br>Scorcerer | Z80 | GPC | ASCII | | INT | INT | X | | BASIC,<br>FORTR.,<br>APL,<br>COBOL | | | | | | 1 | | | X | | | | 16K-ROM-Kassetten einschiebbar, Grafik |
| **Fairchild** | | | | | | | | | | | | | | | | | | | | | | |
| F8-Kit | 3850 | KIT | | | | | | | | 1K | | | | | | | | | | | | |
| BLAZE-16 | 9440 | ES | HEX | | INT | INT | | | | | | | | | 1 | | | | | | | Floppy-INT, PROM-PROGR |
| Mini Formulator | 3850 | GPC | ASCII | | X | | | | | 16K | | | | | | X | | | | | S-100 | Floppy-INT |
| OCM-1 | 3850 | SBC | | | | | | X | ASS | 1K | 2K | | 11K | 64K | | | X | X | | | | |
| SPARK-16 | 9440 | SBC | INT | | INT | INT | | | ASS,<br>BASIC | 4K | 2K | | 24K | 32K | | | X | X | | | | Floppy-INT |
| **Ferranti** | | | | | | | | | | | | | | | | | | | | | | |
| F100 | F100L | SBC | | | | | | | ASS | 8K | | | 64K | | | | | | | | | |
| **Futuredata** | | | | | | | | | | | | | | | | | | | | | | |
| (Neumüller) | | | | | | | | | | | | | | | | | | | | | | |
| AMDS | 6800<br>8080<br>Z80 | ES | ASCII | | X | INT | X | | ASS | 48K | | | | | | | X | X | | | | Doppel-Floppy, Logik-Analysator |
| NET3 | 6800<br>8080<br>Z80 | ES | ASCII | | X | X | X | | ASS,<br>BASIC | 39K | | | | | | | | | | | | Doppel-Floppy, PROM-PROGR |
| **General Instruments** | | | | | | | | | | | | | | | | | | | | | | |
| GIMINI | 1600 | ES | | | | | | | | | | | | 64K | | | | | | | | DMA |
| SC-1600 | 1600 | SBC | | | | | | | | 16K | | 4K | | | | | X | | | | | |
| **HAL** | | | | | | | | | | | | | | | | | | | | | | |
| 8000 | 8080 | GPC | ASCII | | INT | | | X | | 2K | | | | | 1 | | | | | | S-100 | Floppy-INT |
| MCEM-8080 | 8080 | GPC | ASCII | | INT | | | X | | 9K | | | | | 1 | | | | | | | |
| **HARRIS** | | | | | | | | | | | | | | | | | | | | | | |
| (KONTRON)<br>MICRO-12 | 6100 | GPC | HEX | 8 | | | | 1K | | 0,25K | 1K | | 2K | | 1 | | X | X | | | | |
| **Heath Comp.** | | | | | | | | | | | | | | | | | | | | | | |
| 2-H11 | LSI-11 | KIT | ASCII | | X | | X | X | | 8K | | | | | | X | | | | | | Floppy-INT |
| 3-H11 | LSI-11 | KIT | ASCII | | X | | X | X | | 4K | | | | | 1 | | | | | | | " |
| **Heathkit** | | | | | | | | | | | | | | | | | | | | | | |
| ET-3400 | 6800 | KIT | HEX | 6 | | | | 1K | | 0,25K | | | 1,5K | | 1 | | | | | | | |
| H8 | 8080 | LS | HEX | 9 | | | X | X | | 4K | | | 16K | | | | X | X | | | S-100 | |
| H11 | LSI-11 | GPC | INT | | INT | INT | X | | | 8K | | | | 64K | | | X | X | | | | div. Peripherie |

| Hersteller (Distributor) Typ | µP | Art | Tastatur | Anzeige (Stellen) | Bildschirm | Drucker | Netzteil | Monitor (byte) | Sprachen | Speicher -ausbau (byte) RAM | ROM | PROM | direkt | mit Zusatz | Kassettenrecorder Anschlüsse | eingebaut | Schnittstellen V 24 | 20 mA | TTL | IEC | Systembus | Besonderheiten |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **● Hewlett-Packard** | | | | | | | | | | | | | | | | | | | | | | |
| 2108MK | 1000M | GPC | INT | | INT | INT | | | ASS, BASIC, FORTR. | 16K | 1K | | 1M | | | X | X | X | | | HP | Floppy- und Platten-INT |
| 2109EK | 1000E | GPC | INT | | INT | INT | | | " | 16K | 1K | | 1M | | | X | X | X | | | HP | " |
| Lab5036A | 8085 | LS | HEX | 6 | | | X | X | | 1K | 2K | | | | | | | | X | | | |
| **● HPS** | | | | | | | | | | | | | | | | | | | | | | |
| 3517D | 8080 | LS | HEX | 6 | | | X | X | | 0,25K | 1K | | | 64K | 1 | | | | X | | | |
| **● ICSD** | | | | | | | | | | | | | | | | | | | | | | |
| MTS | 8080 | LS | HEX | 8 | | | | 1K | | 1K | | 1K | | | 1 | | | | | | | DMA |
| 525A | 8080 | LS | HEX | 8 | | | X | 1K | | 2K | | 1K | 12K | | 1 | | | | | | S-100 | |
| **● ICT** (Neumüller) | | | | | | | | | | | | | | | | | | | | | | |
| ESCO | 6502 | EBC | INT | 6 | INT | INT | | 2K | ASS | 2K | 8K | | | 64K | 1 | | | X | | | ESCO | DMA |
| **● IMSAI** (Micro Mail) | | | | | | | | | | | | | | | | | | | | | | |
| 8048-CC | 8048 | LS | HEX | | | | | | | 1K | 1K | | 6K | 64K | | | | X | | | SPEZI | |
| IMSAI-8080 | 8080 | BLC | INT | | | | X | | BASIC | | | | | 64K | 1 | | X | X | | | S-100 | Floppy-INT |
| IMSAI-8048 | 8048 | LS | HEX | | | | | | | 1K | | | | | | | | | | | | |
| MPU-8 | 8085 | SBC | | | | | | X | | 0,25K | 2K | | 6K | 64K | | | | | | | S-100 | |
| PCS-80 | 8085 | GPC | ASCII | | X | | X | X | | 16K | | | | | 1 | | | | | | S-100 | Floppy |
| VDP-80 | 8085 | GPC | ASCII | | X | | X | X | BASIC, FORTR., COBOL | 32K | | | | 196K | 1 | | | | | | S-100 | Floppy |
| **● Infinite** | | | | | | | | | | | | | | | | | | | | | | |
| UC1800 | 1802 | KIT | HEX | 6 | INT | | X | | | 0,25K | | | | | 1 | | | | | | S-100 | |
| **● Information Control** | | | | | | | | | | | | | | | | | | | | | | |
| MCS | Z80 | LS | INT | | INT | INT | | | ASS, BASIC | | 1K | | | 64K | 1 | | X | X | | | S-100 | Floppy-INT |
| **● Intel** (Electronic 2000 Jermyn, Neye) | | | | | | | | | | | | | | | | | | | | | | |
| Intellec 40 | 4040 | GPC | BIN | LED | | | | X | ASS | 1,25K | | 8K | | | | | | | | | | PROM-PROGR. |
| Intellec 8 | 8008 | GPC | BIN | LED | | | | X | ASS, PL/M | 16K | | | | | | | | | | | | " |
| Intellec 80 | 8080 | GPC | BIN | LED | | | | X | ASS, PL/M | | 64K | | | | | | | | | | | " |
| LSBC86/12 | 8086 | SBC | INT | | INT | INT | | | ASS, PL/M | 16K | | | 40K | 512K | | | X | X | | | MULTI | Floppy-INT |
| MDS-210 | 8080 | ES | | | | INT | | | ASS, BASIC, FORTR, PL/M | 32K | | | | | | | | | | | | Doppel-Floppy |
| MDS231 | 8080 8085 8048 8086 | ES | | | | | | | | | | | | | | | | | | | | |
| MDS-800 | 8080 | ES | INT | | INT | INT | | X | ASS | 16K | 2K | | | | | | X | X | | | MULTI | DMA |
| SBC80P | 8080 | KIT | INT | | INT | INT | | 2K | | 1K | | 2K | 10K | | | | X | X | | | | SBC80/10 µC |
| SBC80/04 | 8085 | SBC | INT | | INT | INT | | | ASS | 0,25K | | | 4K | | | | X | X | | | | |
| SBC80/05 | 8085 | SBC | INT | | INT | INT | | | ASS | 0,5K | | | 8,5K | 64K | | | X | X | | | MULTI | DMA |
| SBC80/10A | 8080 | SBC | INT | | INT | INT | | | ASS | 1K | | | 10K | 64K | | | X | X | | | MULTI | DMA |
| SBC80/20 | 8080 | SBC | INT | | INT | INT | | | ASS | 2K | | | 14K | 64K | | | X | X | | | MULTI | DMA |
| SBC-86/12 | 8086 | SBC | INT | | INT | INT | | | BASIC, ASS, PL/M | 32K | | | 48K | 1M | | | X | | | | MULTI | |

| Hersteller (Distributor) Typ | µP | Art | Tastatur | Anzeige (Stellen) | Bildschirm | Drucker | Netzteil | Monitor (byte) | Sprachen | Speicher -ausbau (byte) RAM | ROM | PROM | direkt | mit Zusatz | Kassettenrecorder Anschlüsse | eingebaut | Schnittstellen V 24 | 20 mA | TTL | IEC | Systembus | Besonderheiten |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| noch | | | | | | | | | | | | | | | | | | | | | | |
| ● **Intel** | | | | | | | | | | | | | | | | | | | | | | |
| SDK-85 | 8085 | SBC | HEX | 6 | | | | | | 0,5K | 2K | | 7K | 64K | | | X | X | | | | Experimentierfeld |
| SDK-86 | 8086 | SBC | HEX | 6 | INT | | | | ASS, PL/M | 4K | 8K | | 20K | 1M | | | X | X | | | MULTI | " |
| Intellec MDS | 3000<br>8048<br>8080<br>8085<br>8086 | ES | ASCII | | X | INT | | | | | | | | | | | | | | | | Floppy |
| ● **Intelligent Systems** | | | | | | | | | | | | | | | | | | | | | | |
| Intecolor 8001 | 8080 | GPC | ASCII | | X | INT | X | | | 4K | | | | | | | X | X | | | | Floppy-INT |
| Intecolor 8051 | 8080 | GPC | ASCII | | X | INT | X | | | 27K | | | | | 1 | | | | | | S-100 | Floppy |
| ● **Intersil** | | | | | | | | | | | | | | | | | | | | | | |
| Intercept I | 6100 | LS | OCT | 8 | | INT | | | ASS | 0,25K | 1K | | 5K | 32K | 1 | | X | X | | | | Batterieversorgung |
| Intercept II | 6100 | ES | BIN | LED | | INT | X | | | 4K | | | 32K | | | | | X | | | | |
| Intercept Jr. | 6100 | LS | OCT | 8 | | INT | | | ASS | 0,25K | 1K | | 5K | 32K | 1 | | X | X | | | | vgl. Int. I |
| ● **ITT** | | | | | | | | | | | | | | | | | | | | | | |
| MP | 8080 | LS | BIN | LED | | | X | 1K | | 0,25K | 1K | | | 64K | | | | | | | | |
| ● **Kontron** | | | | | | | | | | | | | | | | | | | | | | |
| PSI-80 | Z80 | GPC | ASCII | | X | | X | | BASIC, FORTR, PASCAL | | | | 80K | | 1 | | | | | | S-100 | DMA, Grafik, Floppy |
| Z80-KIT | Z80 | KIT | HEX | 6 | | | | 1K | | 0,25K | 1K | 1K | 4K | 64K | 1 | | X | X | | | | |
| ● **Leybold-Heraeus** | | | | | | | | | | | | | | | | | | | | | | |
| 57500 | 8080 | LS | HEX | 6 | | | X | 1K | | 0,25K | 1K | | 2,75K | | | | | | | | | |
| ● **Luxor** | | | | | | | | | | | | | | | | | | | | | | |
| ABC80 | Z80 | GPC | ASCII | | X | INT | X | | BASIC | 16K | 1,6K | | | | | 1 | X | | | | | Floppy-INT |
| ● **Martin Research** | | | | | | | | | | | | | | | | | | | | | | |
| Mike III | 8080 | LS | INT | LED | INT | | X | X | | 4K | | | | | 1 | | | X | | | S-100 | |
| Mike-8 | Z80 | LS | HEX | LED | | | X | X | | 4K | | | | | | | | X | | | | |
| ● **MCS** | | | | | | | | | | | | | | | | | | | | | | |
| ALPHA 1 | 6502 | LS | HEX | 8 | | | X | 2K | | 1,25K | | | 1K | 64K | 2 | | X | X | X | | MCS | erweiterbar zum BETA |
| BETA 8 | 6502 | BLC | INT | | INT | INT | X | 2K | ASS, BASIC | | | | | 64K | 2 | | X | X | X | X | MCS | EBC |
| ● **MCSF** | | | | | | | | | | | | | | | | | | | | | | |
| EM6800 | 6800 | LS | BIN | LED | | | X | X | | 0,1K | | | | | | | | | | | | |
| ● **MDS** | | | | | | | | | | | | | | | | | | | | | | |
| MD-690 | 6802 | SBC | | | | | | X | ASS | 1K | 2K | | | 56K | | | | | | | S-100 | |
| MD-690A | 6802 | SBC | INT | | INT | INT | | X | ASS, BASIC, PASCAL | 1K | 1K | | 12K | 64K | 1 | | X | | | | S-100 | Floppy-INT |
| F800 | F8 | SBC | INT | | INT | | | X | | 8K | | | | | 1 | | | | | | S-100 | Floppy-INT |
| ● **Microcomp. Assoc.** (Astronic, Neumüller) | | | | | | | | | | | | | | | | | | | | | | |
| Jolt | 6502 | BLC | INT | | INT | INT | | 1K | ASS, BASIC | 0,5K | 1K | | | 64K | 1 | | X | X | | | | Floppy-INT |

| Hersteller (Distributor) Typ | µP | Art | Tastatur | Anzeige (Stellen) | Bildschirm | Drucker | Netzteil | Monitor (byte) | Sprachen | Speicher RAM | Speicher ROM | Speicher PROM | -ausbau (byte) direkt | -ausbau (byte) mit Zusatz | Kassetten-recorder Anschlüsse | Kassetten-recorder eingebaut | Schnitt-stellen V 24 | Schnitt-stellen 20 mA | Schnitt-stellen TTL | Schnitt-stellen IEC | Systembus | Besonderheiten |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Microdata** | | | | | | | | | | | | | | | | | | | | | | |
| MICROCAT | Z80 | GPC | ASCII | | INT | INT | X | | ASS, BASIC, FORTR., COBOL | 8K | | | 48K | | 1 | | X | | | | | Floppy-INT |
| CAT-50 | Z80 | GPC | INT | | INT | INT | X | | " | 32K | | | 64K | | 1 | | X | X | | | | Doppel-Floppy |
| CAT-100 | Z80 | GPC | INT | | INT | INT | X | | " | 32K | | | 64K | 1024K | 1 | | X | X | | | | Doppel-Floppy, Platte |
| **Microproducts** | | | | | | | | | | | | | | | | | | | | | | |
| Superkim | 6502 | SBC | HEX | 6 | | | X | 2K | | 4K | 2K | 16K | | 64K | 1 | | X | X | | | | KIM-kompatibel |
| **Microkit** | | | | | | | | | | | | | | | | | | | | | | |
| 8/16 | 8080 6800 | GPC | ASCII | | X | | X | X | | 8K | | | | | | X | | | | | | Floppy |
| **Microthek** | | | | | | | | | | | | | | | | | | | | | | |
| M-2 | Z80 | ES | ASCII | | INT | INT | X | | | | | | | | 1 | | | | | | | Floppy-INT |
| M-20 | Z80 | ES | ASCII | | X | INT | X | | | | | | | | 1 | | | | | | | Floppy, PROM-PROGR. |
| M-200 | Z80 | ES | ASCII | | X | INT | X | | | | | | | | | | | | | | | Floppy, Fest-platte, PROM-PROGR. |
| M-2000 | Z80 | ES | | | INT | INT | X | | | | | | | | | | | | | | | Floppy- und Platten-INT |
| **Mikrotec** | | | | | | | | | | | | | | | | | | | | | | |
| MTC116 | 6800 9900 | ES | ASCII | | X | X | X | | | | | | | | | | | | | | | Floppy, PROM-PROGR. |
| **Milertronics** | | | | | | | | | | | | | | | | | | | | | | |
| PDC-102 | 8085 | SBC | INT | | INT | INT | | X | ASS | 0,25K | 0,5K | | 2K | 64K | 1 | | | | | | SPEZI | |
| **MITS** (Data Logic, Digitronic) | | | | | | | | | | | | | | | | | | | | | | |
| Altair 6800 | 6800 | GPC | BIN | LED | INT | INT | X | 1K | ASS, BASIC | 1K | | 1K | | 64K | 1 | | X | X | | | S-100 | Floppy-INT |
| Altair 8080 | 8080 | GPC | BIN | LED | INT | INT | X | 1K | ASS, BASIC | 1K | | 1K | | 64K | 1 | | X | X | | | S-100 | " |
| **Monolithic Systems** | | | | | | | | | | | | | | | | | | | | | | |
| MCS-8001 | Z80 | SBC | | | | | | X | ASS, PASCAL | 4K | | | 24K | 64K | | | X | X | | | MULTI | |
| MCS-8080 | 8080 | GPC | ASCII | LED | INT | | | X | | 4K | | | 24K | 64K | 1 | | | | | | S-100 | Floppy-INT |
| **MORAT** | | | | | | | | | | | | | | | | | | | | | | |
| TV-Computer | 6800 8080 SC/MP | LS | | | INT | | X | | | 1K | 1K | | | | 1 | | | | | | | Lichtgriffel, Relays |
| **MOS** | | | | | | | | | | | | | | | | | | | | | | |
| KIM-1 | 6502 | LS | HEX | 6 | | | | 2K | | 1K | 2K | | | 64K | 1 | | | X | | | | |
| **Mostek** (Altron) | | | | | | | | | | | | | | | | | | | | | | |
| F8 SDB | 3850 | SBC | INT | | | | | | ASS | 8K | | | | | | | | X | | | | |
| MD-SBC1 | Z80 | SBC | | | | | | | | 1K | | | 10K | 64K | | | | | | | | |
| MDX-CPU1 | Z80 | SBC | | | INT | INT | | | ASS, BASIC, FORTR. | 0,25K | | | 4,5K | 64K | | | | | | | | Floppy-INT |
| OEM-80 | Z80 | SBC | | | INT | INT | | | " | 4K | | | 36K | 64K | | | X | X | | | | Floppy-INT |
| SYS80F | 3870 Z80 | ES | | | INT | INT | | | | | | | | | | | | | | | | Floppy, PROM-PROGR. |

| Hersteller (Distributor) Typ | µP | Art | Tastatur | Anzeige (Stellen) | Bildschirm | Drucker | Netzteil | Monitor (byte) | Sprachen | Speicher RAM | Speicher ROM | Speicher PROM | -ausbau (byte) direkt | -ausbau (byte) mit Zusatz | Kassetten-recorder Anschlüsse | Kassetten-recorder eingebaut | Schnitt-stellen V 24 | Schnitt-stellen 20 mA | Schnitt-stellen TTL | Schnitt-stellen IEC | Systembus | Besonderheiten |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Motorola** | | | | | | | | | | | | | | | | | | | | | | |
| EXORciser | 6800 | ES | INT | | INT | INT | X | | ASS, BASIC, MPL, FORTR. | | | | | 64K | | | X | X | | | EXOR | Floppy- und Platten-INT |
| EXOR term | 6800 | ES | ASCII | | X | INT | X | | " | | | | | 64K | | | X | X | | | EXOR | " |
| KIT II | 6800 | KIT | HEX | 6 | | | | 1K | | 0,25K | 1K | | 5K | | 1 | | X | X | | | EXOR | |
| MEK6800D2 | 6800 | KIT | HEX | 6 | | | | 1K | | 0,25K | 1K | 2K | 3,5K | 64K | | | | | | | EXOR | |
| M68MM01 | 6800 | SBC | INT | | INT | INT | | | | 1K | | | 6K | 64K | 1 | | | | | | EXOR | Floppy-INT |
| M68MM01A | 6800 | SBC | INT | | INT | INT | | | | 1K | | | 9K | 64K | 1 | | | | | | EXOR | |
| M68MM01B | 6800 | SBC | INT | | INT | INT | | | | 0,12K | | | 8K | 42K | | | | | | | EXOR | |
| M68MM01B1 | 6800 | SBC | INT | | | | | | | 0,3K | | | 8K | 64K | 1 | | X | | | | EXOR | |
| M68PDS | 6800 | ES | ASCII | LED | INT | INT | | | | | | | | 64K | 1 | | | | | | EXOR | Floppy-INT |
| M68TDS | 6800 | ES | ASCII | | X | INT | | | ASS, BASIC | 16K | | | | 64K | 1 | | | | | | EXOR | PROM-PROGR. |
| **Mytek** | | | | | | | | | | | | | | | | | | | | | | |
| Scampi | 8060 | LS | OCT | LED | | | X | | | 0,25K | | | | 64K | | | | | | | | |
| **NAS** (Computershop) | | | | | | | | | | | | | | | | | | | | | | |
| NASCOM1 | Z80 | KIT | ASCII | | INT | | | 1K | | 2K | 1K | | 4K | 16K | 1 | | X | X | | | | |
| **National** | | | | | | | | | | | | | | | | | | | | | | |
| BLC80/10 | 8080 | SBC | INT | | INT | INT | | | ASS, BASIC, FORTR. | 1K | | | 6K | 64K | | | X | X | | | MULTI | Floppy-INT |
| BLC80/11 | 8080 | SBC | INT | | INT | INT | | | " | 1K | | | 9K | 64K | | | X | X | | | MULTI | " |
| BLC80/12 | 8080 | SBC | INT | | INT | INT | | | " | 2K | | | 10K | 64K | | | X | X | | | " | " |
| BLC80/14 | 8080 | SBC | INT | | INT | INT | | | " | 4K | | | 12K | 64K | | | X | X | | | " | " |
| BLC80/204 | 8080 | SBC | INT | | INT | INT | | | " | 4K | | | 16K | 64K | | | X | | | | " | " |
| IMP-16C | IMP-16 | SBC | | | | | | | | 1K | | | 2K | 128K | | | | | | | | DMA |
| IMP-16L | IMP-16 | SBC | INT | | INT | | | X | | 8K | | | | | 1 | | | | | | | |
| ISP-8C/100E | SC/MP | EBC | | | | | | | | 0,25K | | | 0,75K | 64K | | | | | | | | DMA |
| PACE | PACE | SBC | INT | | INT | INT | | | | 1K | | | | | | | | | | | | Floppy-INT |
| SC/MP-KIT | SC/MP | KIT | | | | | | 0,5K | | 0,25K | 0,5K | | | 64K | | | X | | | | | |
| SC/MP-LCDS | SC/MP | LS | HEX | 6 | | | | | | 0,25K | 0,5K | | | 6K | | | | X | | | | |
| SC/MP-EUROCARDS | SC/MP | BLC | INT | | INT | INT | | | | 0,25K | | | 0,75K | 64K | 1 | | | | | | | EBC, ADC |
| STARPLEX | 8080 | ES | ASCII | | X | X | X | | ASS, BASIC, FORTR. | 64K | | | | | | | X | | | | | Doppel-Floppy, PROM-PROGR. |
| **NEC** (Microscan) | | | | | | | | | | | | | | | | | | | | | | |
| TK80 | 8080 | LS | HEX | 8 | | | | 0,75K | | 0,25K | | 0,75K | 2K | 64K | | | | | X | | | DMA |
| **North Star** (Digitronic) | | | | | | | | | | | | | | | | | | | | | | |
| Horizon | Z80 | GPC | INT | | INT | | X | X | BASIC | 16K | | | | 64K | | | X | X | | | S-100 | Doppel-Floppy |
| **Ohio Scientific** (Bisy) | | | | | | | | | | | | | | | | | | | | | | |
| Challenger | 6502 | GPC | ASCII | | X | | X | X | BASIC | 12K | 10K | | 32K | | | | X | | | | | Floppy |
| 6502SBC | 6502 | GPC | ASCII | | INT | | | | BASIC, ASS | 8K | 8K | | 40K | | 1 | | | | | | | Floppy-INT, Grafik |
| **Omnibyte** | | | | | | | | | | | | | | | | | | | | | | |
| OB8001 | 6800 | SBC | INT | | INT | INT | | | ASS, BASIC | 1,1K | | | 6,25K | 64K | 1 | | X | X | | | | Floppy-INT |
| **Pacific Cyber** | | | | | | | | | | | | | | | | | | | | | | |
| PCM-12A | 6100 | KIT | INT | | INT | | | X | | 1K | | | | | 1 | | | | | | S-100 | Floppy-INT |
| **Pagel** | | | | | | | | | | | | | | | | | | | | | | |
| µCS-178 | 8080 | GPC | HEX | | INT | INT | X | 3K | | 1K | | | 10K | | 1 | | X | X | | | | |
| **Panafacom** | | | | | | | | | | | | | | | | | | | | | | |
| LKIT-16 | | KIT | HEX | LED | INT | INT | | 1K | BASIC | 0,5K | 1K | | 3K | 64K | 1 | | | X | | | | |

| Hersteller (Distributor) Typ | µP | Art | Tastatur | Anzeige (Stellen) | Bildschirm | Drucker | Netzteil | Monitor (byte) | Sprachen | Speicher RAM | Speicher ROM | Speicher PROM | -ausbau (byte) direkt | -ausbau (byte) mit Zusatz | Kassetten-recorder Anschlüsse | Kassetten-recorder eingebaut | Schnitt-stellen V 24 | Schnitt-stellen 20 mA | Schnitt-stellen TTL | Schnitt-stellen IEC | Systembus | Besonderheiten |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **PCS** | | | | | | | | | | | | | | | | | | | | | | |
| PCS S II | LSI-11 | ES | INT | | INT | INT | | | BASIC FORTR. | 8K | | | | 32K | 1 | | | | | | | Floppy, Platte |
| **PEP** | | | | | | | | | | | | | | | | | | | | | | |
| µpep6408 | 6800 | BLC | INT | | INT | INT | | | | 0,25K | | | 1K | 64K | | | X | X | | | IEC | EBC, ADC, Floppy-INT |
| PSM6800 | 6800 | SBC | | | | INT | | | | 4K | | 4K | | 64K | | | | | | | EXOR | |
| **Pertec** | | | | | | | | | | | | | | | | | | | | | | |
| 680b | 6800 | SBC | | | | | X | | ASS, BASIC | 1K | 0,25K | | 2K | 64K | | | | | | | | |
| **Philips** | | | | | | | | | | | | | | | | | | | | | | |
| Instructor 50 | | | | | | | | | | | | | | | | | | | | | | siehe **VALVO** |
| **Plessey** | | | | | | | | | | | | | | | | | | | | | | |
| Miproc16 | Miproc16 | BLC | INT | | INT | | X | X | ASS | 16K | | | | 128K | 1 | | | | | | | Floppy-INT |
| Miproc PK | Miproc16 | LS | HEX | | | | X | X | | 0,5K | 1K | | | | | | | | | | | Lochstreifen-leser |
| **Polymorphic** (Digitronic) | | | | | | | | | | | | | | | | | | | | | | |
| POLY-88 | 8080 | GPC | ASCII | | INT | | X | | ASS, BASIC | 16K | 11K | 1K | | | 1 | | | | | | S-100 | |
| **Process Computer Systems** | | | | | | | | | | | | | | | | | | | | | | |
| 1810A | 8080 | SBC | INT | | INT | INT | | | BASIC, ASS | 1K | 3K | | 8K | 64K | | | X | X | | | | Floppy-INT |
| 1880A | Z80 | SBC | INT | | INT | INT | | | " | 1K | 6K | | 14K | 64K | | | X | X | | | | " |
| **Processor Technology** | | | | | | | | | | | | | | | | | | | | | | |
| SOL-20 | 8080 | GPC | ASCII | | INT | | X | X | BASIC | 2K | | 1K | 4K | 64K | 1 | | X | X | | | S-100 | |
| SOL-PC | 8080 | SBC | INT | | INT | INT | | | ASS, BASIC | 2K | | | 4K | 64K | 1 | | X | X | | | S-100 | Floppy-INT |
| **Pro Didaktik** | | | | | | | | | | | | | | | | | | | | | | |
| MPU40104 | | LS | BIN | 2 | | | X | | | 0,5K | | | | | | | | | | | | |
| **Pro-log** (Spezial-El.) | | | | | | | | | | | | | | | | | | | | | | |
| PLS-858 | 8085 | SBC | | | | | | | | 1K | | 10K | 64K | | | | | | | | | |
| PLS-868 | 6800 | SBC | | | | | | | | 1K | | 10K | 64K | | | | | | | | | |
| PLS-898 | Z80 | SBC | | | | | | | | 1K | | 10K | 64K | | | | | | | | | |
| **Quay** (Digitronic) | | | | | | | | | | | | | | | | | | | | | | |
| Q80AI | Z80 | GPC | INT | | | | | 1K | BASIC | 1K | | 1K | 5K | 64K | | | X | X | | | | PROM-PROGR. |
| Q90MPS | Z80 | SBC | INT | | INT | INT | | | BASIC | 4K | | 1K | 65K | 80K | 1 | | X | X | | | | " |
| Q90F/MPS | Z80 | SBC | INT | | INT | INT | | | BASIC | 4K | | 1K | 65K | 80K | | | X | X | | | | Floppy |
| **Radio Shack** | | | | | | | | | | | | | | | | | | | | | | siehe **Tandy** |
| **RCA** (Neye, SASCO) | | | | | | | | | | | | | | | | | | | | | | |
| COSMAC CDP18S005 | 1802 | ES | INT | 6 | INT | INT | X | | ASS | 4K | | | | 64K | 1 | | X | X | | | | Floppy-INT |
| COSMAC CDP18S012 | 1802 | LS | INT | | | | X | | | 0,25K | | | | | | | X | X | | | | |
| COSMAC CDP18S020 | 1802 | KIT | | | | | | X | | 0,25K | | | 4K | | | | X | X | | | | |
| VIP | 1802 | KIT | HEX | | INT | | | 0,5K | | 2K | 0,5K | | 4K | 64K | 1 | | | | | | | TV-Spiele |

| Hersteller (Distributor) Typ | µP | Art | Tastatur | Anzeige (Stellen) | Bildschirm | Drucker | Netzteil | Monitor (byte) | Sprachen | Speicher RAM | Speicher ROM | Speicher PROM | -ausbau (byte) direkt | -ausbau (byte) mit Zusatz | Kassetten-recorder Anschlüsse | Kassetten-recorder eingebaut | Schnitt-stellen V 24 | Schnitt-stellen 20 mA | Schnitt-stellen TTL | Schnitt-stellen IEC | Systembus | Besonderheiten |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Rockwell** | | | | | | | | | | | | | | | | | | | | | | |
| AIM65 | 6502 | SBC | ASCII | 20 | | X | | 8K | ASS | 1K | 8K | | 32K | 64K | 2 | | | X | | | | |
| PPS MP | PPS-4 | ES | HEX | | | | X | | | 2K | | 4K | | 32K | | | X | X | | | | Floppy-INT, |
| | PPS-8 | | | | | | | | | | | | | | | | | | | | | Lochstreifen |
| SYS65 | 6502 | ES | INT | | INT | INT | | 16K | ASS | 16K | 16K | | | | | | X | X | | | | Dual-Floppy |
| XPO-1 | PPS-4 | LS | HEX | 5 | | | | X | | 1K | 2K | | 2K | | | | X | X | | | | |
| **Roederstein** | | | | | | | | | | | | | | | | | | | | | | |
| TK-80 | | | | | | | | | | | | | | | | | | | | | | vgl. **NEC** |
| EZ80DIT | 8080 | LS | HEX, BIN | 8 | | | X | 1K | | 0,5K | 1K | | 2K | | 1 | | | | | | | |
| **Salota** | | | | | | | | | | | | | | | | | | | | | | |
| MFC512 | Z80 | GPC | INT | | INT | INT | X | | ASS, BASIC | | | | 64K | | 1 | | X | X | | | S-100 | Floppy |
| **SD-Systems** | | | | | | | | | | | | | | | | | | | | | | |
| SBC-100 | Z80 | SBC | INT | | INT | | | | | 1K | | | 9K | 64K | | | X | | | | S-100 | Floppy-INT |
| **SGS-Ates** | | | | | | | | | | | | | | | | | | | | | | |
| Z80-Nano | Z80 | SBC | HEX | 8 | INT | | | | ASS | 4K | | | 8K | | 2 | | X | X | | | | |
| Z80-Micro | Z80 | SBC | ASCII | | | | | X | ASS | 4K | 16K | 4K | 32K | 64K | 2 | | X | X | | | | Floppy-INT |
| **SHARP** | | | | | | | | | | | | | | | | | | | | | | |
| BAMZ-80K | Z80 | GPC | ASCII | | X | INT | X | | BASIC | 20K | 4K | | 48K | | | X | | | | | | |
| **Siemens** | | | | | | | | | | | | | | | | | | | | | | |
| 210D | 8080 | BLC | | | | | | | | 1K | | | 5K | 64K | | | X | X | | | | |
| 210E | 8080 | BLC | | | | | | | | 1K | | | 5K | 64K | | | X | X | | | | EBC |
| AMC29 | 2900 | ES | INT | | INT | INT | X | | | | | | | 64K | | | X | | | | | Floppy-INT, PROM-PROGR. |
| AMC95/4000 | 9080 | SBC | | | | | | | | | | | | | | | | | | | | siehe **AMC** |
| ECB85 | 8085 | LS | | | | | | | | | | | | | | | | | | | | " |
| Mikroset 8080 | 8080 | LS | HEX | 6 | | | X | 1K | | 0,5K | 1K | | 2K | 64K | 1 | | | | | | | |
| PC-100 | 6502 | GPC | ASCII | 20 | INT | X | X | 8K | BASIC | 4K | 16K | | | 64K | 2 | | | X | | X | | ADC |
| SBC-8010 | 8080 | SBC | | | | | | 2K | | 1K | | | 5K | | | | X | X | | | | |
| SBC-8020 | 8080 | SBC | | | | | | 2K | | 2K | | | 6K | | | | X | | | | | |
| SIKIT-DK/8080 | 8080 | KIT | INT | | | | | 1K | | 0,25K | | 1K | 5K | | | | | X | | | | |
| SKC85 | 8085 | EBC | | | | | | | | | | | | | | | | X | | | | siehe AMC |
| SME800 | 8080 | ES | BIN | LED | INT | | X | | ASS | 16K | 2K | 0,25K | | 64K | | | | X | | | | Floppy-INT |
| SMP80 | 8080 | BLC | | | | | | | | | | | | | | | | | | | | siehe AMC |
| SYS-8010 | 8080 | GPC | INT | | INT | | X | 2K | | 1K | | | 16K | | | | X | X | | | | |
| **Signetics** | | | | | | | | | | | | | | | | | | | | | | s. VALVO |
| **SKS/EMK** | | | | | | | | | | | | | | | | | | | | | | |
| Kiss | 8085 | GPC | ASCII | | INT | INT | X | 4K | BASIC | 1K | 32K | | | 64K | 1 | | X | X | | | | Floppy-INT |
| **Southwest** (Carroll & Co) | | | | | | | | | | | | | | | | | | | | | | |
| SWTP6800 | 6800 | GPC | INT | | INT | | X | X | | 4K | | | | | 1 | | X | X | | | | |
| SWTP68/2 | 6800 | GPC | ASCII | | X | | X | 1K | ASS, BASIC | 32K | 1K | | 40K | 64K | | | X | X | | | S-50 | |
| **Steiner** (Astronic) | | | | | | | | | | | | | | | | | | | | | | |
| PUZZLE | 6502 | BLC | INT | | INT | INT | | 11K | ASS | 0,5K | | 2K | | 64K | 1 | | X | X | | | | EBC |
| **Stoppani** | | | | | | | | | | | | | | | | | | | | | | |
| DELPHIN CLUB | 2650 | KIT | OCT | 4 | INT | | | 0,25K | | 0,25K | 0,25K | | 0,75K | | 1 | | | | | | | Batterien |
| DELPHIN INDUSTRIE | Z80 | SBC | OCT | 4 | INT | INT | X | 0,25K | BASIC | 1,25K | 0,25K | | 3,75K | | 1 | | X | X | | | | PROM-PROGR. |
| **Synertek** (Electronic 2000) | | | | | | | | | | | | | | | | | | | | | | |
| CP110 | 6502 | SBC | INT | | | | | 1K | ASS, BASIC | 1K | 1K | | 6K | 64K | | | X | X | X | | | Super Jolt |

| Hersteller (Distributor) Typ | µP | Art | Tastatur | Anzeige (Stellen) | Bildschirm | Drucker | Netzteil | Monitor (byte) | Sprachen | Speicher RAM | Speicher ROM | Speicher PROM | -ausbau (byte) direkt | -ausbau (byte) mit Zusatz | Kassetten-recorder Anschlüsse | Kassetten-recorder eingebaut | Schnitt-stellen V 24 | Schnitt-stellen 20 mA | Schnitt-stellen TTL | Schnitt-stellen IEC | Systembus | Besonderheiten |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| noch | | | | | | | | | | | | | | | | | | | | | | |
| • **Synertek** | | | | | | | | | | | | | | | | | | | | | | |
| SM100 | 6502 | SBC | INT | | INT | | | 4K | ASS, BASIC | 1K | 4K | | 32K | 64K | | | X | X | X | | | OEM-Version von SYM-1 |
| SYM-1 | 6502 | LS | HEX | 6 | INT | | | 4K | | 1K | 4K | | 32K | 64K | 1 | | X | X | X | | | |
| VIM-1 | 6502 | SBC | | | | | | | | | | | | | | | | | | | | |
| • **Tandy** (Radio Shack) | | | | | | | | | | | | | | | | | | | | | | |
| TRS-80 | Z80 | GPC | ASCII | | X | INT | X | | BASIC | 4K | | | | 48K | 1 | | X | X | | | | Floppy-INT |
| • **TANO** | | | | | | | | | | | | | | | | | | | | | | |
| OUTPOST11 | 6800 | GPC | ASCII | | X | INT | X | | BASIC | 32K | | | 64K | | | | X | | | | | Floppy |
| • **Technical Design Labs** (Digitronic) | | | | | | | | | | | | | | | | | | | | | | |
| XITAN | Z80 | GPC | INT | | INT | | | 2K | ASS, BASIC, FORTR. | 2K | 2K | | | 64K | 1 | | | | | | S-100 | Floppy-INT |
| • **Tektronix** (Rohde & Schwarz) | | | | | | | | | | | | | | | | | | | | | | |
| 8002 | 8080 6800 Z80 9900 F8 | ES | ASCII | | X | X | X | | ASS | 16K | 16K | | | 64K | | | X | | | | | Floppy |
| • **Texas** | | | | | | | | | | | | | | | | | | | | | | |
| AMPL | 9900 9940 9980 | ES | ASCII | | X | INT | X | | FORTR. | 24K | | | | | | | | | | | | Doppel-Floppy |
| CS990/4 | 9900 | ES | ASCII | | INT | X | X | X | ASS | 4K | | | | | 2 | | | | | | | |
| LCM-1001 | TMS1000 | LS | BIN | | LED | | X | | | 0,5K | | | 1,5K | | | | | | X | | | Batterien |
| TI990/4 | 9900 | ES | BIN | | LED | INT | X | X | ASS | | | | 32K | | 1 | | | | | | | Floppy-INT, PROM-PROGR. |
| TM990/100M | 9900 | SBC | INT | | | INT | | | ASS | 0,25K | | 1K | 6K | 32K | 1 | | X | X | | | | |
| TM990/101M | 9900 | SBC | INT | | | INT | | | ASS | 2K | | 1K | 11K | 32K | 1 | | X | X | | | | |
| TM990/180M | 9980 | SBC | INT | | | INT | | | ASS | 0,5K | | 2K | 7,5K | | 1 | | X | X | | | | |
| • **Toyo** | | | | | | | | | | | | | | | | | | | | | | |
| Pecker-1 | Z80 | LS | HEX | 16 | | | X | | | 16K | | | 23K | | | | | X | | | | EPROM-Programmer |
| • **Valvo-Signetics** | | | | | | | | | | | | | | | | | | | | | | |
| DS2000 | 2650 | LS | INT | LED | | | X | X | | 5K | 1K | | | 32K | | | X | X | | | | DMA |
| Instructor 50 | 2650 | LS | HEX | 8 | | | X | 2K | | 0,5K | | | | 32K | 1 | | | | | | S-100 | |
| KT100SK | 8 x 300 | KIT | | | | | | | | 0,25K | | 0,5K | | | | | | | | | | |
| KT9500 | 2650 | KIT | | LED | | | | X | | 0,5K | 1K | | | 32K | | | X | X | | | | DMA |
| MIKIT2650 | 2650 | KIT | HEX | 6 | | | | 1K | | 1K | | | | | 1 | | X | | | | | |
| PC1001 | 2650 | SBC | | LED | | | | X | | 1K | 1K | | | 32K | | | X | X | | | | DMA |
| PC1500 | 2650 | SBC | | LED | | | | X | | 0,5K | 1K | | | 32K | | | X | X | | | | DMA |
| TWIN | 2650 | ES | ASCII | | X | INT | X | | | 16K | 16K | | | | 1 | | X | X | | | | PROM-PROGR., Doppel-Floppy |
| VA200 | 2650 | EBC | | | | INT | | | | 0,25K | | 4K | | | | | | X | X | | | |
| • **Vector Graphic** (Digitronic) | | | | | | | | | | | | | | | | | | | | | | |
| VECTOR1A | 8080 | GPC | INT | | INT | | X | 0,5K | | 1K | | 0,5K | 3,5K | | 1 | | | | | | S-100 | Floppy-INT |
| VECTOR1AZ | Z80 | GPC | INT | | INT | | X | 0,5K | | 1K | | 0,5K | 3,5K | | 1 | | | | | | S-100 | " |
| VECTOR1+ | 8080 Z80 | GPC | INT | | INT | | X | 0,5K | | 1K | | 0,5K | 3,5K | | 1 | | | | | | S-100 | Floppy |

| Hersteller (Distributor) Typ | µP | Art | Tastatur | Anzeige (Stellen) | Bildschirm | Drucker | Netzteil | Monitor (byte) | Sprachen | Speicher -ausbau (byte) RAM | ROM | PROM | direkt | mit Zusatz | Kassettenrecorder Anschlüsse | eingebaut | Schnittstellen V 24 | 20 mA | TTL | IEC | Systembus | Besonderheiten |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| • **Wave Mate** | | | | | | | | | | | | | | | | | | | | | | |
| Jupiter II C | 6800 | GPC | ASCII | | X | | X | X | | 8K | | | | | | X | X | X | | | | Floppy-INT |
| Jupiter III C | Z80 | GPC | ASCII | | X | | X | X | | 8K | | | | | | X | X | X | | | | " |
| • **Western Data** | | | | | | | | | | | | | | | | | | | | | | |
| DATA Handler | 6502 | LS | BIN | LED | | | | X | | 1K | | | | | | | | X | | | S-100 | |
| • **Western Digital** (Spezial-Electr.) | | | | | | | | | | | | | | | | | | | | | | |
| MC-160 | 1600 | BLC | | | | | | | | 0,5K | 1K | | 10K | 64K | | | X | X | | | | DMA, Floppy-INT |
| • **Zilog** (Kontron) | | | | | | | | | | | | | | | | | | | | | | |
| Z80-DS1 | Z80 | ES | | | | | X | 4K | ASS | 16K | 3K | | | 36K | | | X | X | | | | |
| Z80-DSU | Z80 | ES | | | | INT | X | 4K | ASS | 16K | 3K | | | 64K | | | X | X | | | | Floppy-INT, PROM-PROGR. |
| Z80-ECB | Z80 | BLC | | | | | | | | 0,25K | | 1K | | 64K | | | X | X | | | | EBC |
| Z80-KIT | Z80 | KIT | HEX | 6 | | | | 1K | | 0,25K | | 1K | | 64K | 1 | | X | X | | | | |
| Z80-MCB | Z80 | BLC | INT | | INT | INT | | X | | 4K | | | 20K | 64K | | | X | X | | | | |
| Z80-MCS | Z80 | ES | INT | | INT | INT | X | X | ASS, BASIC | 16K | | 3K | | 64K | | | X | X | | | | Doppel-Floppy |
| Z80-MCZ | Z80 | ES | INT | | INT | INT | X | X | ASS | 16K | | 3K | | 64K | | | X | X | | | | Doppel-Floppy |
| Z80-PDS | Z80 | ES | INT | | INT | | X | X | ASS | 16K | | 3K | | | | | X | X | | | | |


# 65xx MICRO MAG

## COMPUTING · SOFTWARE · HOBBY

2-monatliche deutsche Fachzeitschrift für 65xx- Mikroprozessoren wie AIM 65, PC100, PET, KIM-1, OSI, APPLE, SYM-1 usw.

mit vielen wertvollen und nachvollziehbaren Programmen in jeder Ausgabe. Es handelt sich grundsätzlich um Erstveröffentlichungen in deutscher Sprache, die nur im 65xx MICRO MAG erscheinen. Besonderes Augenmerk wird auf verständliche Kommentierung und auf Programme gelegt, die die Systementwicklung erleichtern. Geschickte Programmierungstechniken werden hervorgehoben und mit Beispielen belegt. Berichte über Anwenderschaltungen und Literaturhinweise ergänzen die Darstellungen.

65xx MICRO MAG hat mit seinen bis August 1979 vorliegenden 8 Ausgaben eine besonders anerkennende Aufnahme in Fachkreisen gefunden (Firmen, Hochschulinstitute, ernsthaft interessierte private Betreiber). Überzeugen bitte auch Sie sich mit einem Probeexemplar.

Abonnement für 6 Ausgaben ab laufender Nr. DM 40,– (Ausland/Foreign DM 46,–).
Die früheren Hefte sind zu je DM 7,– nachlieferbar.

Weiter im Lieferprogramm: Thermorollen für AIM 65/PC100 in kontrastreicher Spitzenqualität, AIM 65 Anwenderhandbuch in deutscher Sprache.

**HERAUSGEBER/BEZUG: DIPL.-VOLKSWIRT ROLAND LÖHR, HANSDORFER STRASSE 4**
**D-2070 AHRENSBURG, ☎ 04102-55816**

# 2 Adressen

Nicht selten wird die zügige Erledigung einer Aufgabe dadurch behindert, daß die richtige Adresse fehlt. Ein Anlaß könnte der Wunsch sein, beim Hersteller direkt Antwort auf ein grundsätzliches Problem einzuholen. Oder man möchte wissen, wer ein gesuchtes Gerät vertreibt. Bei Kaufentscheidungen wird vielleicht von Interesse sein, ob ein ausländischer Hersteller in der Nähe eine Agentur (einen *Distributor*) hat. Denn im Garantiefall oder im allgemeinen Schadensfall sollte eine schnelle Reparatur oder ein Umtausch möglich werden. Schließlich würde mancher gerne die praktischen Erfahrungen und Programmsammlungen eines Anwender- oder Benutzerclubs nutzen wollen oder sich in Lehrinstituten angemessene Nachhilfen erteilen lassen, wenn nur die Anschriften bekannt wären.

Mit dieser ersten Ausgabe des Jahrbuches ist nun ein Anfang gemacht, um bei solchen und ähnlichen Nöten für Abhilfe zu sorgen. Die Hersteller sind in der Regel mit ihrer Originaladresse angegeben. Soweit bekannt, ist zusätzlich die deutsche Tochterfirma oder ein Repräsentant aufgeführt. Die vollständige Adresse des Repräsentanten oder Distributors kann dem Abschnitt mit den Lieferantenadressen entnommen werden.

Die Lieferantenadressen sind ergänzt durch die Angabe des in unserem Zusammenhang relevanten Hauptproduktes. Bei den Lehrinstituten ist stichwortartig genannt, was zu den Themen Kleinrechner, Speicher und Programmierung geboten wird.

## 2.1 Taschen- und Tischrechner-Hersteller

**ADDIATOR**
Rechenmaschinenfabrik
C. Kübler
Postfach 1168
7620 Wolfach/Baden

**ARISTO**
Postfach 50 0380
2000 Hamburg 50

**Aviatronik**
Rosenheimer Landstraße 39
8012 Ottobrunn

**BMC, Business Machine Corporation**
Osaka 540, Japan
in Deutschland:
Unitrans GmbH

**BOHSEI Enterprise Co.**
22-1 Shinsen-cho Shibuya-ku
Tokyo 150, Japan
und
20 501 Plummer Street
Chatsworth
Calif. 91311, USA

**BRAUN AG**
Postfach 1120
6242 Kronberg/Taunus

**BROTHER Int. GmbH**
Postfach 1320
6368 Bad Vilbel

**BRUNSVIGA**
(s. Olympia)

**Canon Inc.**
11-28 Mita 3-chome
Minato-ku
Tokyo 108, Japan
in Deutschland:
Gustav A. Mücher
Bürotechnik

**CASIO**
2-6 Nishi-Shinjuku,
Shinjuku-ku
Tokyo, Japan
in Deutschland:
Casio GmbH
Kieler Straße 212
2000 Hamburg 54

**CBM**
Mareno, P. Blümel & Co
Georgstraße 20
3000 Hannover 1
(s. auch Commodore)

**CI**
(s. ITOH)

**CITIZEN Business Mach. Inc.**
1-48 Okubo I-chome,
Shinjuku-ku
Tokyo, Japan
in Deutschland:
ESSELTE CITIZEN
Büromaschinen GmbH
Kestnerstraße 42
3000 Hannover 1

**COLEX**
M. u. F. Herrnböck KG
Pappenheimstraße 7
8000 München 2

**Commodore Business Mach., Inc.**
901 California Avenue
Palo Alto
Cal. 94304, USA
in Deutschland:
Commodore GmbH
Frankfurter Str. 171–175
6078 Neu-Isenburg

**Compucorp**
Los Angeles, Cal., USA
in Deutschland:
Beaugrand Datentechnik GmbH
Niederröderweg 30
6056 Heusenstamm

**COMTEC Corporation**
9-1 Shinjuku 1-chome,
Shinjuku-ku
Tokyo, Japan

**ELECTROGAMMA S. A.**
Case postale 199
CH-1207 Genf 3, Schweiz

**A. W. Faber-Castell**
8504 Stein bei Nürnberg

**FACIT GmbH**
Postfach 130 260
4000 Düsseldorf 13

**General Corp. Japan**
1116 Suenaga, Takatsu-ku
Kawasaki 213, Japan
in Deutschland:
KANEMATSU-GOSHO GmbH
Karl-Rudolf-Str.178
4000 Düsseldorf

**GUTAG AG**
CH-8958 Spreitenbach 2,
Schweiz

**HANIMEX GmbH**
Postfach 1644
3012 Langenhagen 1

**HEATHKIT GmbH**
Postfach 102060
6072 Dreieich-Sprendlingen

**HELEC Hunte electronic GmbH**
Am Sportplatz 35–37
2901 Huntlosen

**Hermes**
(s. PRECISA)

**Hewlett-Packard GmbH**
Postfach 560 140
6000 Frankfurt 56

**Hitachi Ltd.**
Tokyo, Japan
in Deutschland:
Königsallee 6
4000 Düsseldorf

**IBICO GmbH**
Stettiner Ring 11
6382 Friedrichsdorf 4

**INTERTON Electronic**
Postfach 860 230
5000 Köln 80

**C. ITOH & Co, LTD.**
C.P.O. Box 136
Tokyo, Japan

**Jet-Electronics GmbH**
Danckelmannstraße 24
1000 Berlin 19
(s. auch Sabtronics)

**Keystone GmbH**
Sulzbacher Str. 29–39
6231 Schwalbach

**KOVAC Corp.**
C.P.O. Box 1319
Osaka, Japan
in Deutschland:
Unitrans GmbH

**LITRONIX, Inc.**
19000 Homestead Road
Cupertino, Cal. 95014, USA
in Deutschland:
LITRONIX GmbH
Postfach 710 328
6000 Frankfurt 71

**Litton Royal**
Royal-Imperial Int.
Lyoner Straße 11
6000 Frankfurt 71

**LOGITECH Inc.**
56 Nanking E. Rd., Sec. 4
Taipei, Taiwan

**MARENO**
(s. CBM)

**Matsushita Electric**
Panasonic, Japan
in Deutschland:
Statronic

**MBO**
Schmidt & Niederleitner GmbH
Kowad-Celtis-Str. 79
8000 München 70

**NASCO Corp.**
1-13, 4-chome, Kudan-Kita,
Chiyoda-ku
Tokyo, Japan

**National Semicond. Corp.**
2900 Semiconductor Drive
Santa Clara, Cal. 95051, USA
in Deutschland:
Industriestraße 10
8080 Fürstenfeldbruck

**NIKKO & Co., Ltd.**
2-9-1, Nishi Shinbashi,
Minato-ku
Tokyo 105, Japan

**Nissin Enterprise Corp.**
2-15-11, Takaido-Higashi,
Suginami-ku
Tokyo 168, Japan

**Deutsche Olivetti GmbH**
Lyoner Str. 34
6000 Frankfurt 71

**Olympia Werke AG**
Postfach 960
2940 Wilhelmshaven

**OMRON TATEISI Electr. Co.**
6-1-1, Heiwajima, Ohta-ku
Tokyo 143, Japan
in Deutschland:
OMRON Europe GmbH
Hamburger Str. 11
2000 Hamburg 76

**PRECISA**
Hermes Precisa GmbH
Postfach 5780
6236 Eschborn/Taunus

**Radofin Electronic GmbH**
Ramlinger Str. 50
3167 Burgdorf-Ehlershausen

**Realtone AG**
Alte Winterthurerstr. 53
CH-8304 Wallisellen, Schweiz

**RENO**
(s. MARENO)

**Rockwell International**
Anita House,
Rockingham Road
Uxbridge, Middlesex,
UB82XL, England
in Deutschland:
Industriestraße 8
6374 Steinbach/Taunus

**ROYAL IMPERIAL**
(s. TRIUMPH ADLER)

**Sabtronics International**
Winkelriedstraße 35
CH-6003 Luzern, Schweiz

**SANYO**
Electric Trading Co., Ltd.,
Japan
in Deutschland:
SANYO Deutschland GmbH
Widenmayerstr. 25
8000 München 22

**SERD**
(s. SANYO)

**SHARP Corp.**
22-22 Nagaike-cho, Abeno-ku
Osaka, Japan
in Deutschland:
SHARP ELECTR. GmbH
Steindamm 11
2000 Hamburg 1

**SILVER SEIKO INT. GmbH**
Lyoner Str. 44–48
6000 Frankfurt 71

**Sinclair Radionics Ltd.**
London Road
St. Ives, Huntingdonshire,
England
in Deutschland:
Sinclair Elektronik GmbH
Rosenheimer Landstraße 39
8012 Ottobrunn

**Suntie Electronics Co, Ltd.**
Japan

**Superlectron B.V.**
P.O. Box 16
Vianen, Holland

**Systema AG**
P.O. Box 34684
F.L. 9490 Vaduz Im. Lett
Fürstentum Liechtenstein

**TEALTRONIC**
Büromaschinen GmbH
Peckhauser Str. 55
4020 Mettmann II

**Texas Instruments Deutschland GmbH**
Haggertystraße 1
8050 Freising

**TOSHIBA Corp.**
Tokyo, Japan
in Deutschland:
Hammer Landstr. 115
4040 Neuss 1

**TOSHO Co., Ltd.**
7-2 Ohtemachi 1-chome,
Chiyoda-ku
Tokyo, Japan

**TOWA SANKIDEN CORP.**
4-2 Goban-cho, Chiyoda-ku
Tokyo, Japan

**TRICOM Corp.**
13-6 Ebisu 1-chome,
Shibuya-ku
Tokyo 150, Japan

**Triumph-Adler Vertriebs GmbH**
Fürther Str. 212
8500 Nürnberg

**VICTOR**
Walter Kidde & Comp.
Ninoofsesteenweg 71
1750 Schepdaal, Belgien

**WALTHER Büromaschinen GmbH**
Postfach 15
7921 Gerstetten

## 2.2 Mikroprozessoren-Hersteller

**AMD, Advanced Micro Devices**
901 Thompson Place
Sunnyvale, Cal. 94085, USA
in Deutschland:
Rosenheimer Straße 139
8000 München 80

**AMI, American Microsystems, Inc.**
3800 Homestead Road
Santa Clara, Cal. 95051, USA
in Deutschland:
AMI Microsystems GmbH
Rosenheimer Straße 30
8000 München 80

**Data General Corp.**
Route 9
Southboro, Mass. 01772, USA
in Deutschland:
Data General GmbH
Breitlacherstraße 96
6000 Frankfurt

**EMM Electronic Memories**
12621 Chadron Avenue
Hawthorne, Cal. 90250, USA
in Deutschland:
EMM GmbH
Hauffstraße 4
8000 München 70

**Eurosil GmbH**
Weltenburgerstraße 6
8000 München 80

**Fairchild**
464 Ellis Street
Mountain View, Cal. 94042, USA
in Deutschland:
Fairchild Halbleiter GmbH
Daimlerstraße 15
8046 Garching-Hochbrück

**Ferranti Electric**
E. Bethpage Rd.
Plainview, New York 11803, USA
in Deutschland:
Ferranti GmbH
Widenmayerstr. 5
8000 München 22

**Fujitsu Ltd.**
6-1 Marunochi 2-chome,
Chiyoda-ku
Tokyo, Japan
in Deutschland:
Elektron System Herborg KG
oder Logitec

**General Instrument Corp.**
600 West John Street
Hicksville, New York 11802, USA
in Deutschland:
General Instrument GmbH
Nordenstraße 1a
8000 München 40

**HARRIS Semiconductor**
P.O. Box 883
Melbourne, Fl. 32901, USA
in Deutschland:
KONTRON GmbH

**Hitachi Ltd.**
2700 River Road
Des Plaines, Il. 60018, USA
in Deutschland:
Königsallee 6
4000 Düsseldorf

**Hughes Microelectronics**
500 Superior Ave.
Newport Beach, Cal. 92663, USA

**Intel Corporation**
3065 Bowers Avenue
Santa Clara, Cal. 95051, USA
in Deutschland:
Intel Semiconductor GmbH
Seidlstraße 27
8000 München 2

**INTERSIL Inc.**
10900 N. Tantau Ave.
Cupertino, Cal. 95014, USA
in Deutschland:
SE Spezial-Electronic KG

**ITT Semiconductors**
74 Commerce Way
Woburn, Mass. 01801, USA
in Deutschland:
ITT INTERMETALL GmbH
7800 Freiburg

**MMI Monolithic Memories Inc.**
1165 East Arques Avenue
Sunnyvale, Cal. 94086, USA
in Deutschland:
Neumüller GmbH

**MOS Technology, Inc.**
950 Rittenhouse Rd.
Norristown, Pa. 19401, USA
in Deutschland:
Commodore
und Neumüller

**MOSTEK**
1215 West Crosby Road
Carollton, Texas 75006, USA
in Deutschland:
MOSTEK GmbH
Talstraße 172
7024 Filderstadt 1

**MOTOROLA Semiconductor Inc.**
P.O. Box 20912
Phoenix, Ariz. 85036, USA
in Deutschland:
Motorola GmbH
Münchner Straße 18
8043 Unterföhring

**National Semiconductor Corp.**
2900 Semiconductor Drive
Santa Clara, Cal. 95051, USA
in Deutschland:
Eisensheimerstr. 61/2
8000 München 21

**NEC, Nippon Electric Company**
Japan
in Deutschland:
MS Microscan

**PANAFACOM, Ltd.**
17-15 Shinbashi, 6-chome,
Minato-ku
Tokyo, Japan

**Panasonic**
Matsushita Electric
Japan
in Deutschland:
Statronic

**Plessey Microsystems**
1674 McGaw Ave.
Irvine, Cal. 92714, USA
in Deutschland:
Plessey GmbH
Motorstraße 56
8000 München 40

**Raytheon Semiconductor**
350 Ellis Street
Mountain View, Cal. 94042, USA
in Deutschland:
CARROLL & Co.

**RCA Solid State Division**
P.O. Box 3200
Somerville, New Jersey 08876, USA
in Deutschland:
RCA GmbH
Justus-von-Liebig-Ring 10
2085 Quickborn

**Rockwell International Corp.**
P.O. Box 3669
Anaheim, Cal. 92803, USA
in Deutschland:
Rockwell Int. GmbH
Fraunhoferstraße 11
8033 München-Martinsried

**Scientific Micro Systems**
520 Clyde Ave.
Mt. View, Cal. 94043, USA

**Sescosem**
(s. Thomson-CSF)

**SGS-ATES**
79 Massosoit-Street
Waltham, Mass. 02154, USA
in Deutschland:
Haidling 17
8018 Grafing

**Siemens AG**
Balanstraße 73
8000 München 80

**Signetics**
811 East Arques Ave.
Sunnyvale, Cal. 94086, USA
in Deutschland:
s. VALVO-Signetics

**Solid State Scientific Inc.**
Montgomeryvale, Pa. 18936, USA
in Deutschland:
NEUTRON GmbH

**Synertek**
3050 Coronado Drive
Santa Clara, Cal. 95051, USA
in Deutschland:
Astronic, Bitronic, Electronic 2000

**Telefunken**
AEG-Telefunken
Theresienstraße 2
7100 Heilbronn

**Texas Instruments**
P.O. Box 1444
Houston, Texas 77001, USA
in Deutschland:
TI Deutschland GmbH
Haggertystraße 1
8050 Freising

**Thomson-CSF**
38, rue Vauthier
92100 Boulogne-Billancourt, Frankreich
in Deutschland:
Fallstraße 42
8000 München 70

**TOSHIBA Corp.**
Tokyo, Japan
in Deutschland:
Hammer Landstraße 115
4040 Neuss 1

**VALVO-Signetics**
Burchardstraße 19
2000 Hamburg 1

**Western Digital Corp.**
3128 Redhill Ave.
Newport Beach, Cal. 92663, USA
in Deutschland:
SE Spezial-Electronic

**Zilog**
10460 Bubb Road
Cupertino, Cal. 95014, USA
in Deutschland:
Zugspitzstraße 4
8011 Vaterstetten

## 2.3 Mikrocomputer-Hersteller

**ACORN Computer**
in Deutschland:
BETZ Elektronik-Vertrieb

**AMC**
Advanced Micro Computers GmbH
(s. Siemens AG)

**AMD, Advanced Micro Devices**
Rosenheimer Straße 139
8000 München 80

**AMI Microsystems GmbH**
Rosenheimer Straße 30
8000 München 80

**Apple Computer, Inc.**
20863 Stevens Creek Blvd.
Cupertino, Cal. 95014, USA
in Deutschland:
BASIS Microcomputer Vertrieb GmbH

**ATARI, Inc.**
1265 Borregas Ave.
Sunnyvale, Cal. 94086, USA

**Applied Systems**
100 N. Winchester Blvd.
Santa Clara, Cal. 95050, USA

**Canon Inc.**
11-28 Mita 3-chome Minato-ku
Tokyo 108, Japan
in Deutschland:
Gustav A. Mücher Bürotechnik

**Christiani**
Technisches Lehrinstitut
Postfach 1710
7750 Konstanz

**Commodore GmbH**
Frankfurter Straße 171–175
6078 Neu-Isenburg

**COMPAS**
Computer Application Corp., USA
in der Schweiz:
Erhardt + Jost Electronik

**Compucolor Corp.**
5965 Peachtree Corners East
Norcross, Ga. 30071, USA
in der Schweiz:
Derungs AG

**Compucorp**
1901 S. Bundy Dr.
Los Angeles, Cal. 90025, USA
in Deutschland:
Beaugrand Datentechnik GmbH
Niederröderweg 30
6056 Heusenstamm

**Computer Automation**
NAKED MINI Division
18651 Von Karman
Irvine, Cal. 92713, USA
in Deutschland:
Computer-Automation GmbH
Otto-Hahn-Straße 23
6072 Dreieich

**Control Logic Corp.**
Nine Tech. Circle
Natick, MA 01760, USA

**Cramer Electronics**
85 Wells Ave.
Newton, Mass. 02159, USA

**Cromemco**
2400 Charleston Rd.
Mountain View, Cal. 94043, USA
in Deutschland:
Digitronic Computersysteme GmbH

**Data General GmbH**
Breitlacherstraße 96
6000 Frankfurt

**DATARAM Corp.**
Princeton-Hightstown Rd.
Cranburg, NI 08512, USA

**Dietz Computer Systeme**
Solinger Straße 9
4330 Mühlheim-Ruhr

**Digital Electronics Corp.**
2126 Sixth St.
Berkeley, Cal. 94710, USA

**Digital Equipment Corp. (DEC)**
Parker St.
Maynard, Mass. 01754, USA
in Deutschland:
Digital Equipment GmbH
Wallensteinplatz 2
8000 München 40

**Digital Group**
P.O. Box 6528
Denver, Col. 80206, USA

**Digitronic Computersysteme GmbH**
Bei der Doppeleiche 3–5
2000 Wedel

**EDS Vertriebsgesellschaft mbH**
Bodenseestraße 235
8000 München 60

**EKF Elektronik Meßtechnik**
Am Pilsholz 4
4700 Hamm 1

**E & L Instruments, Inc.**
61 First St.
Derby, CT 06418, USA

**ELTEC Elektronik GmbH**
Neubrunnenstraße 10
6500 Mainz

**EMM GmbH**
Hauffstraße 4
8000 München 70

**Exidy Sorcerer**
in Deutschland:
MICRO COSMOS GmbH

**Fairchild Halbleiter GmbH**
Daimlerstraße 15
8046 Garching-Hochbrück

**Ferranti GmbH**
Widenmayerstraße 5
8000 München 22

**Futuredata**
11205 S. La Cienega Blvd.
Los Angeles, Cal. 90045, USA
in Deutschland:
Neumüller Meßtechnik

**General Instrument GmbH**
Nordenstraße 1a
8000 München 40

**HAL Communications**
807 E. Green Street
Urbana, Ill. 61801, USA

**HARRIS Semiconductor**
Box 883
Melbourne, FL 32901, USA
in Deutschland:
KONTRON GmbH

**Heath Comp.**
Benton Harbor, MI 49022, USA

**Heathkit GmbH**
Robert-Bosch-Straße 32–38
6072 Dreieich-Sprendlingen

**Hewlett-Packard GmbH**
Postfach 560 140
6000 Frankfurt 56

**HPS System-Technik**
Postfach 661
4300 Essen 1

**ICSD Integrated Computer Systems Deutschland**
Leonrodstraße 58
8000 München 19

**ICT**
(s. ESCO)

**IMSAI, IMS Associates, Inc.**
14860 Wicks Blvd.
San Leandro,Cal. 94577, USA
in Deutschland:
Micro Mail

**Infinite, Inc.**
1924 Waverly Place
Melbourne, Fla. 32901, USA

**Intel Semiconductor GmbH**
Seidlstraße 27
8000 München 2

**Intelligent Systems Corp.**
5965 Peachtree Corners East
Norcross, GA 30071, USA

**INTERSIL, Inc.**
711 Stierlin Rd.
Mt. View. Cal. 94043, USA
in Deutschland:
SE Spezial-Electronic KG

**ITT INTERMETALL GmbH**
7800 Freiburg

**KONTRON Elektronik GmbH**
Breslauer Straße 2
8057 Eching bei München

**Leybold-Heraeus GmbH**
Postfach 51 0760
5000 Köln 51

**Luxor Electronic GmbH**
Kieler Straße 555
2000 Hamburg 54

**Martin Research**
336 Commercial Ave.
Northbrook, Ill. 60062, USA

**MCS Micronic Computer Systeme GmbH**
Zentralflughafen Tempelhof
1000 Berlin 42

**MCSF Michael Franzki**
Adlerring 16
7730 VS-Villingen

**MDS Microdata Systems**
2 Mack Road
Woburn, Mass. 01801, USA
in Deutschland:
MDS-Deutschland GmbH
Oskar-Jäger-Straße 175
5000 Köln 30

**Microcomputer Assoc., Inc.**
2859 Scott Blvd.
Santa Clara, Cal. 95050, USA
in Deutschland:
Neumüller GmbH

**Microdata**
17481 Red Hill Ave.
Irvine, Cal. 92705, USA
in Deutschland:
Wallbrunnstraße 15
7850 Lörrach

**Microkit, Inc.**
2180 Colorado Ave.
Santa Monica, Cal. 90404, USA

**Microproducts**
2107 Artesia Blvd.
Redondo Beach, Cal. 90278, USA

**Microthek Datensysteme**
Börsenstraße 17
6000 Frankfurt 1

**Microtec**
Box 60 337
Sunnyvale, Cal. 94088, USA

**MITS (Altair)**
2450 Alamo S. E.
Albuquerque, NM 87106, USA
in Deutschland:
Data Logic und Digitronic

**Monolithic Systems Corp.**
14 Inverness Dr. East
Englewood, Colo. 80110, USA

**Franz MORAT KG**
7821 Eisenbach 1

**MOS Technology, Inc.**
950 Rittenhouse Rd.
Norristown, Pa. 19401, USA
in Deutschland:
Commodore und Neumüller

**MOSTEK GmbH**
Talstraße 172
7024 Filderstadt 1

**Motorola GmbH**
Münchner Straße 18
8043 Unterföhring

**Mytek Mikro-Elektronik GmbH**
Kaiserstraße 160 A
4600 Dortmund

**NAS-Elektronik**
Briennerstraße 56
8000 München 2

**National Semiconductor**
Eisenheimerstraße 61/2
8000 München 21

**National Matsushita Electric**
in Deutschland:
Statronic

**NEC, Nippon Electric Company**
in Deutschland:
MS Microscan

**North Star**
2547 Ninth St.
Berkeley, Cal. 94710, USA
in Deutschland:
Computershop und Digitronic

**Ohio Scientific Instr.**
11679 Hayden St.
Hiram, Ohio 44234, USA
in Deutschland:
Bisy

**Omnibyte Corp.**
2711-B Curtiss St.
Downers Grove, Ill. 60515, USA

**Pacific Cyber/Metrix, Inc.**
180 Thorup Lane
San Ramon, Cal. 94583, USA

**Hartmut Pagel**
Microcomputertechnik
Elmblick 33
3300 Braunschweig

**PANAFACOM, Ltd.**
17-15 Shinbashi, 6-chome,
Minato-ku
Tokyo, Japan

**PCS**
Pfälzer-Wald-Straße 36
8000 München 90

**PEP Elektronik Systeme GmbH**
Gutenbergstraße 9B
8950 Kaufbeuren

**Pertec Computer Corporation**
20630 Nordhoff St.
Chatsworth, Cal. 91311, USA

**Philips**
(s. VALVO-Signetics)

**Plessey GmbH**
Motorstraße 56
8000 München 40

**Polymorphic Systems**
737 S. Kellog
Galeta, Cal. 94608, USA
in Deutschland:
Digitronic

**Process Computer Systems Inc.**
750 North Maple Road
Saline, Mich. 48176, USA

**Processor Technology Corp.**
7100 Johnson Industrial Drive
Pleasanton, Cal. 94566, USA

**Pro Didaktik GmbH**
Damaschkestraße 12
4030 Ratingen

**PRO-LOG Corp.**
2411 Garden Rd.
Monterey, Cal. 93940, USA
in Deutschland:
SE Spezial-Electronic

**QUAY Corp.**
Box 386
Freehold, NJ 07728, USA
in Deutschland:
Digitronic

**Radio Shack**
2617 West 7th St.
Fort Worth, Texas 76017, USA
(s. Tandy)

**RCA GmbH**
Justus-von-Liebig-Ring 10
2085 Quickborn

**Rockwell Int. GmbH**
Fraunhoferstraße 11
8033 München-Martinsried

**ROEDERSTEIN GmbH, Ditratherm**
Ludmillastraße 23–25
8300 Landshut

**Salota**
Nürnberg

**SD Systems**
P.O. Box 28810
Dallas, Texas 75228, USA

**SGS-ATES**
Haidling 17
8018 Grafing

**SHARP Electr. GmbH**
Steindamm 11
2000 Hamburg 1

**Siemens AG**
Balanstraße 73
8000 München 80

**Signetics**
S. VALVO-Signetics

**SKS/EMK**
S.6.37–38
6800 Mannheim 1

**Southwest Technical Products (SWTPC)**
219 W. Rhapsody
San Antonio, Texas 78216, USA
in Deutschland:
COI Systeme München
CARROLL & Co.

**Ing. Ernst Steiner**
Geylinggasse 16
A-1130 Wien, Österreich

**STOPPANI AG**
Könizstraße 29
CH-Bern 21, Schweiz

**Synertek**
3050 Coronado Drive
Santa Clara, Cal. 95051, USA
in Deutschland:
Astronic, Bitronic, Electronic 2000

**Tandy Radio Shack**
Immermannstraße 57
4000 Düsseldorf 1

**TANO Corp.**
4521 W. Napoleon Ave.
Metairie, La. 70001, USA

**Technical Design Labs**
Research Park, Bldg. H
1101 State Road
Princeton, N. J. 08540, USA
in Deutschland:
Digitronic

**Tektronix, Inc.**
P.O. Box 500
Beaverton, Or. 97005, USA
in Deutschland:
Rohde & Schwarz GmbH
Sedanstraße 13–17
5000 Köln 1

**Texas Instruments Deutschland GmbH**
Haggertystraße 1
8050 Freising

**Toyo**
in Deutschland:
Electronica GmbH

**VALVO-Signetics**
Burchardstraße 19
2000 Hamburg 1

**Vector Graphic, Inc.**
717 Lakefield Rd.
Westlake, Cal. 91361, USA
in Deutschland:
Digitronic

**Wave Mate**
1015 W. 190 th Street
Gardena, Cal. 90248, USA

**Western Data Systems**
3650 Charles St.
Santa Clara, Cal. 95050, USA

**Western Digital Corp.**
3128 Redhill Ave.
Newport Beach, Cal. 92663, USA
in Deutschland:
SE Spezial-Electronic

**Zilog**
10460 Bubb Road
Cupertino, Cal. 95014, USA
in Deutschland:
Zugspitzstraße 4
8011 Vaterstetten

## 2.4 Lieferanten (Distributoren)

**ABC Computer-Shop GmbH**
Schellingstraße 33
8000 München 40
*(KIM, KISS, NAS, North Star)*

**Adcomp GmbH**
Horemansstraße 8
8000 München 19
*(Drucker)*

**Altron**
Vertrieb Elektronischer Bauelemente
Germaniastraße 10
3160 Lehrte
*(AMD, Mostek)*

**Apel Büro-Center GmbH**
Heinrich-Büssing-Ring 40
3300 Braunschweig
*(Canon, Commodore)*

**ARNOLD-Elektronik**
6362 Wöllstadt 1
*(Altair, IMSAI, Software, Zubehör)*

**Astronic**
Winzererstraße 47d
8000 München 40
*(KIM, Microcomp. Assoc., MMI, Steiner, Synertek)*

**ATLANTIK Elektronik GmbH**
Hofmannstraße 20
8000 München 70
*(SMC, Zubehör)*

**Balü Electronic**
Burchardplatz1
2000 Hamburg 1
*(ICSD)*

**BASIS Microcomputer Vertrieb GmbH**
Friedrich-Ebert-Straße 137
4400 Münster
*(Apple)*

**BETZ Elektronik Vertrieb**
Münchner Straße 7
8023 Pullach
*(ACORN)*

**Bisy Beratungsgesellschaft für Informationssysteme mbH**
Postfach 47
4040 Neuss 1
*(Ohio Scientific)*

**Bitronic GmbH**
Einsteinstraße 127
8000 München 80
*(Synertek, Software)*

**Bussmann-Computerbase**
Mühlenstraße 3
4790 Paderborn
*(Apple, IMSAI, PET, Zubehör)*

**BYTE SHOP LOGITEC**
Sonnenstraße 25
8000 München 2
*(Bausätze, Peripherie, Software, Zubehör)*

**CARROLL & Co. GmbH**
Burnitzstraße 34
6000 Frankfurt 70
*(ITT, Monsanto, Mostek, NEC, Raytheon, RCA, SWTPC)*

**CELDIS GmbH**
Henschelring 5
8011 Kirchheim
*(LSI-11)*

**COI Systeme München**
Arabellastraße 13–15
8000 München 81
*(SWTPC)*

**Computershop GmbH**
Unterortstraße 10
6336 Eschborn
*(NASCOM, North Star)*

**COMTEC GmbH**
Widenmayerstraße 1
8000 München 22
*(FUJITSU)*

**Conrac GmbH**
Industriestraße 18
6992 Weikerskeim
*(Peripherie)*

**Cosmos Elektronik GmbH**
Hegelstraße 16
8000 München 83
*(AMD)*

**DACOS Software GmbH**
Neue Bahnhofstraße 21
6670 St. Ingbert
*(Software, IMSAI, Vector Graphic)*

**Dahms-Elektronik GmbH**
Bürgermeister-Neff-Str. 19
6806 Viernheim
*(NASCOM)*

**Data Logic Computer GmbH**
Im Sechholder 11
6230 Frankfurt 80
*(MITS)*

**DATA MEGA KG**
8011 Putzbrunn bei München
*(Bausätze, Peripherie)*

**DEMA-Elektronik GmbH**
Blütenstraße 21
8000 München 40
*(NASCOM)*

**Walter Derungs AG**
Dübendorfstraße 335
CH-8051 Zürich, Schweiz
*(APPLE, COMPUCOLOR, PET)*

**Digitronic Computersysteme GmbH**
Bei der Doppeleiche 3–5
2000 Wedel
*(Cromemco, IMSAI, MITS, North Star, Polymorphic, QUAY, Tandy, Techn. Design Labs, VECTOR GRAPHIC)*

**EBV-Elektronik**
In der Meineworth 9A
3006 Burgwedel 1
*(AMD, Motorola)*

**EHV Elektronik Hobby Versand GmbH**
An der Breiten Wiese 9
3000 Hannover 61
*(Commodore, Elrad-Triton, Bausätze)*

**Elbatex GmbH**
Cäcilienstraße 24
7100 Heilbronn
*(AMD, FUJITSU, KIM)*

**Electric Contact**
Kasernenstraße 3
7300 Esslingen
*(NASCOM)*

**Electric Shop**
Mozartstraße 42
8261 Burgkirchen
*(PET, Texas)*

**ELECTRONIC 2000 Vertriebs GmbH**
Neumarkter Straße 75
8000 München 80
*(Intel, SGS/ATES, Synertek)*

**Electronic GmbH**
Ost-West-Straße 49
2000 Hamburg 11
*(Toyo (Pecker))*

**Electronic-Hobby-Shop**
Kaiserstraße 20
5300 Bonn 1
*(AIM, ALPHA 1, KIM, PET, SYM)*

**Elektronikladen**
Wilhelm-Mellies-Straße 88
4930 Detmold 18
*(Bausätze, Peripherie)*

**Elektron System Herborg KG**
Innerer Ring 1
8302 Mainburg
*(FUJITSU)*

**Elrad Versand**
Verlag Heinz Heise
Postfach 2746
3000 Hannover 1
*(DELPHIN, Bausätze, Fachliteratur)*

**Elsy-Elektronische Systeme GmbH**
Im Rott 14
5180 Eschweiler
*(Schach-Computer)*

**Eltec-Elektronik GmbH**
Neubrunnenstraße 10
6500 Mainz
*(NASCOM)*

**EMK Handelsgesellschaft mbH**
S6. 37–38
6800 Mannheim 1
*(KISS)*

**Erhardt + Jost Electronik**
Rössligasse 2
CH-4450 Sissach, Schweiz
*(Compas)*

**Feltron Elektronik**
Postfach 1169
5210 Troisdorf
*(µC-Information, Fachbücher, Bausätze, Terminals)*

**FEY, Dipl.-Ing. Ernst Fey**
Horemansstraße 28
8000 München 19
*(Data General, DEC, Intel, National)*

**Fuhrmeister & Co**
Ballindamm 17
2000 Hamburg 1
*(Hitachi)*

**Geveke Elektronik GmbH**
Schlägelstraße 9
4000 Düsseldorf 1
*(Peripherie)*

**GWK Technische Elektronik mbH**
Aachener Straße 56
5132 Uebach-Planenberg
*(AIM, KIM, SYM, Software)*

**Heimcomputer Vertriebs GmbH (HOMECOMPUTER)**
Flügelstraße 47
4000 Düsseldorf 1
*(Peripherie)*

**HEW Technik**
Postfach 3108
5810 Witten 3
*(Texas Instr.)*

**Ing. W. Hofacker GmbH**
Tegernseer Straße 18
8150 Holzkirchen
*(Literatur, Software)*

**HPS System Technik**
Postfach 661
4300 Essen 1
*(Lerncomputer)*

**ICS Individuelle Computersysteme GmbH**
Neue Bahnhofstraße 21
6670 St. Ingbert
*(IMSAI, Vector Graphic)*

**Interelektronik**
Dorfstraße 7
2081 Bönningstedt
*(IMSAI, Polymorphic, SWTP, Tandy, Vector Graphic, Xitan, Peripherie)*

**Jermyn GmbH**
Postfach 1146
6277 Camberg
*(Intel, Motorola)*

**KONTRON Elektronik GmbH**
Breslauer Straße 2
8057 Eching bei München
*(Harris, TOSHIBA, Zilog)*

**Logitec GmbH**
Kaiser-Wilhelm-Straße 26
8130 Starnberg
*(FUJITSU)*

**Mareno, P. Blümel & Co.**
Georgstraße 20
3000 Hannover 1
*(CBM, Reno)*

**MCDS Microcomputer Datensysteme GmbH**
Luisenplatz 4
6100 Darmstadt
*(Peripherie, Zubehör)*

**Microcomputer Shop**
Münzenloher GmbH
Tölzerstraße 5
8150 Holzkirchen
*(PET, Bausätze, Software)*

**MICRO COSMOS GmbH**
Kl. Budengasse 1–3
5000 Köln 1
*(Exidy, Horizon, Texas, Peripherie, Software)*

**Microdata GmbH**
Wallbrunnenstraße 15
7850 Lörrach
*(CAT)*

**Micro Logic**
Gesellschaft für rechnergesteuerte Systeme mbH
Tassiloplatz 7
8000 München
*(Interfaces)*

**Micro Mail GmbH**
Gillitzerstraße 1
8200 Rosenheim
*(Cromemco, Digital Group, IMSAI, Northstar, Peripherie)*

**Micro-Shop Bodensee**
M. & R. Nedela
Marktstraße 3
7778 Markdorf
*(KIM, SYM, Bücher, Zeitschriften)*

**MS Microscan**
Schlossgarten 1
8045 Ismaning bei München
*(NEC)*

**Gustav A. Mücher Bürotechnik**
Fraunhoferstraße 14
8033 München-Martinsried
*(Canon)*

**Mütron, Müller & Co. KG**
Bornstraße 22
2800 Bremen 1
*(Motorola)*

**nbn ELEKTRONIK**
Max-Emanuel-Straße 8
8130 Starnberg
*(Peripherie)*

**Neumüller GmbH**
Karlstraße 55
8000 München 2
*(AMI, ESCO, Futuredata, ICT, KIM, Microcomp. Assoc., MMI, MOS, NEC)*

**NEUTRON GmbH**
Postfach 124
6050 Offenbach
*(Solid State, Scientific)*

**Alfred Neye-Enatechnik GmbH**
Schillerstraße 14
2085 Quickborn
*(Data General, Intel, Mostek, Motorola, RCA, Peripherie)*

**Nordelektronik Vertriebs GmbH**
Harksheiderweg 238
2085 Quickborn
*(AMD, MMI)*

**Poly Comp Computer-Systeme GmbH**
Karlstraße 28
7410 Reutlingen 1
*(IMSAI, Vector Graphic)*

**Quelle International**
8510 Fürth 500
*(Privileg)*

**Raffel electronics**
Lochnerstraße 1
4030 Ratingen 1
*(Mostek, Software, Zubehör)*

**Rohde & Schwarz GmbH**
Sedanstraße 13–17
5000 Köln 1
*(Tektronix)*

**RTG E. Springorum GmbH**
Postfach 426
4600 Dortmund 1
*(Motorola)*

**Runow Büroelektronik**
Eisenacher Straße 73
1000 Berlin 62
und
Bachstraße 104
2000 Hamburg 76
*(PET-Zubehör, Speicher, Peripherie)*

**SASCO GmbH**
Hermann-Oberth-Straße 16
8011 Putzbrunn bei München
*(RCA, Motorola)*

**Ing. W. Schwind**
Landsberger Straße 439
8000 München 60
*(Drucker)*

**SE Spezial-Electronic KG**
Ortlerstraße 8
8000 München 70
*(Fairchild, INTERSIL, Pro-log, Western Digital)*

**Spoerle Electronic KG**
Otto-Hahn-Straße 13
6072 Dreieich
*(Bauteile, Motorola)*

**Statronic**
Eppendorfer Weg 244
2000 Hamburg 20
*(Matsushita, Panasonic)*

**STEMMER Elektronik**
Adenauerstraße 9
8031 Puchheim
*(DEC, Intel, Mostek, National, Zilog)*

**SYNELEC Datensysteme GmbH**
Lindwurmstraße 17
8000 München 2
*(Peripherie)*

**SYSTEM-KONTAKT GmbH**
Siemensstraße 5
7107 Bad Friedrichshall
*(LSI-11, Rockwell, Kurse)*

**tdc Data Communications**
Frankfurter Allee 45–47
6236 Eschborn
*(Peripherie)*

**TEP Elektronik Vertriebs GmbH**
Mühlenstieg 9
2000 Hamburg 70
*(FTI 990–30/E)*

**F. Trommeschläger**
Kirchbüchel 1
5330 Königswinter 1
*(diverse µP)*

**Unitrans GmbH**
Postfach 711 042
8000 München 71
*(BMC, Kovac)*

**UNITRONIC**
Münsterstraße 338
4000 Düsseldorf
*(Delphin, EHC-80, Peripherie)*

**VERO GmbH**
Pontstraße 60
5100 Aachen
*(PET, Bausätze, Peripherie)*

**G. Volk Electronics**
Postfach 81 1006
8000 München 81
*(8080, Z-80, Software)*

**Wetronic Automation GmbH**
Claude-Lorrain-Straße 29
8000 München 90
*(Peripherie)*

## 2.5 Clubs und Softwarelieferanten

**Compute, National Semiconductor GmbH**
Industriestraße 10
8080 Fürstenfeldbruck
*(Spezialclub für „National"-Benutzer)*

**Computer Center Dortmund**
H. U. Kampschulte
Emschertalstraße 131
4600 Dortmund 41

**Computer Club Berlin**
Fasanenstraße 67
1000 Berlin 15
*(verschiedene µC-Gruppen)*

**Computer Club Eurex GmbH**
Postfach 1433
7850 Lörrach
*(verschiedene µC-Systeme)*

**Computer Club Europe e.V.**
Postfach 11 0868
6100 Darmstadt
*(verschiedene µC-Systeme)*

**Computer-Club Schweiz**
Alfred Stolz
Lindenstraße 30
CH-9204 Andwil, Schweiz

**Computer-Hobbyclub e.V.**
Marian Jetter
Mozartstraße 13
6233 Kelkheim
*(verschiedene µC-Systeme)*

**EDV-Benutzerverband**
Postfach 29
8473 Pfreimd
*(PET-Programme und Peripherie)*

**Hamburger Computer-Club e.V.**
Günther Giese
Kampstraße 34
2085 Quickborn
*(verschiedene µC-Systeme)*

**Hewlett-Packard Users Club**
Herrenberg Straße 130
7030 Böblingen
*(Programme für BASIC-Tischrechner)*

**Ing. W. Hofacker GmbH**
Tegernseer Straße 18
8150 Holzkirchen
*(Software für PET, 6502, 6800, 8080)*

**Markert-Welfens & Partner GmbH**
Ubierring 25
5000 Köln 1
*(Programme für HP 67, HP 97)*

**µC-User's Club Bitmuster**
M. Reimer
Haubachstraße 8
1000 Berlin 10
oder
G. Kleene
Hasenroth 3
4600 Dortmund 12

**PET-Benutzerklub**
Frankfurter Straße 171–175
6078 Neu-Isenburg
*(Benutzerklub der Commodore GmbH)*

**Schweizer Computer Club**
Seeburgstraße 18
CH-6002 Luzern
*(Software für div. µC und Taschenrechner)*

**Dietmar Severitt**
Gutenbergstraße 20
6052 Mühlheim
*(BASIC-Progr. f. PET, TRS-80 etc.)*

**Sowa-Elektronik Vertriebs-GmbH**
General-von-Stein-Straße 17
8050 Freising
*(div. Programme für TI-59)*

**The Boston Systems Office, Inc.**
D. D. Wallis
Maria-Eich-Straße 11
8032 Lochham
*(Software-Service für viele µC)*

**TI 58/59 Softwareclub**
H. Plewnia
Hintermayrstraße 3
8500 Nürnberg

**TRS-80-Anwender-Club**
Wolfgang Luce
Im Ortfeld 12
4300 Essen 14

## 2.6 Lehrinstitute

**Christiani**
Technisches Lehrinstitut
Dr.-Ing. habil Paul Christiani
Postfach 1672
7750 Konstanz
*(µP-Fernlehrgang)*

**DAG-Technikum**
Kastanienallee 52/54
4300 Essen
*(µP-Fortbildung)*

**DAG-Technikum**
Gutenbergstraße 5
6457 Maintal 2
*(Fernlehrgang Elektronik)*

**Elektronik Lehrinstitut Dirksen GmbH**
Tempelstraße 8
4240 Emmerich 1
*(Kurse mit SDK 85)*

**Handwerkskammer Hildesheim**
Kruppstraße 18
3200 Hildesheim
*(Ganztageskurse µC)*

**Haus der Technik e. V.**
Postfach 767
4300 Essen 1
*(µP-Lehrgänge)*

**HPI, Heinz-Piest-Institut**
Wilhelm-Busch-Straße 18
3000 Hannover 1
*(Fachlehrgang Mikrocomputer)*

**HPS System Technik**
Postfach 661
4300 Essen 1
*(µP-Lehrsystem)*

**CELDIS GmbH**
Henschelring 5
8011 Kirchheim
*(Schulung mit TI 9900)*

**ICSD**
Leonrodstraße 58
8000 München 19
*(Seminare und Fernlehrgang)*

**Institut für Fernunterricht**
Postfach 34 0226
2800 Bremen 34
*(µP-Fernlehrgang)*

**INTEL-Semiconductor GmbH**
Seidlstraße 27
8000 München 2
*(diverse Intel-Seminare)*

**ITT Fachlehrgänge**
Postfach 1570
7530 Pforzheim
*(Fernlehrgänge Digital-Elektronik und µP)*

**Kienzle-Fachschule für Datenverarbeitung**
Postfach 1640
7730 Villingen
*(Einführungskurse)*

**Kontron Elektronik GmbH**
Breslauer Straße 2
8057 Eching
*(Z80-Kurse)*

**MCS Micronic Computer Systeme**
Zentralflughafen Tempelhof
1000 Berlin 42
*(ALPHA 1- und BETA 8-Kurse)*

**Microcomputer-Shop**
Abteilung Seminare
Tölzer Straße 5
8150 Holzkirchen
*(SC/MP- und Z80-Seminare)*

**Alfred Neye-Enatechnik GmbH**
Schillerstraße 14
2085 Quickborn
*(Z80- und COSMAC-Anwenderkurse)*

**N. V. Philips/Elcoma**
Microprocessor Dept.
Building BAE-2
Eindhoven, Niederlande
*(Seminare mit Valvo/Signetics-µP)*

**PSI, Gesellschaft für Prozeßsteuerungs- und Informationssysteme mbH**
Katharinenstraße 19/20
1000 Berlin 31
*(div. Seminare über DV, ORGA, µP, Speicher)*

**R. Schneeberger, µP-Schulung**
Kanalstraße 15
8043 Unterföhring
*(div. Kurse)*

**S. V. I. GmbH**
Spechtweg 2
4150 Krefeld
*(BASIC-Fernlehrgang für PET)*

**SYBEX Europe**
313 Rue Lecourbe
75015 Paris, Frankreich
*(µP-Kurse in ganz Europa)*

**System-Kontakt GmbH**
Siemensstraße 5
7107 Bad Friedrichshall
*(6502-Workshops)*

**Technische Akademie Esslingen**
Postfach 1269
7302 Ostfildern 2
*(verschiedene Kurse und Lehrgänge)*

**Tele-Studio Meier**
Oberascherstraße 12
8000 München 71
*(Übungskurse mit verschiedenen µP)*

**Texas Instruments**
Haggertystraße 1
8050 Freising
*(TI µP-Workshops)*

**VDI-Bildungswerk**
Graf-Recke-Straße 84
4000 Düsseldorf 1
*(µP-Hardware- und-Software-Kurse)*

**VHS Dortmund**
Bornstraße 1
4600 Dortmund
*(Aktivkursus mit CA 8060)*

**VHS Hamm**

*(Taschenrechner und µP in der Erwachsenenbildung)*

# 3 Bücher, Zeitschriften

Vor ein paar Jahren wurde von Technikern und interessierten Laien jede publizistische Neuerscheinung zu den Themen „Taschenrechner" und „Mikroprozessor" mit hohen Erwartungen beachtet und oft auch gelesen. Denn es war schon fast die Regel, daß Neuentwicklungen schneller da waren als brauchbare Literatur dazu. Spezielle Fachzeitschriften gab es natürlich auch nicht gleich. Die Elektronik-Zeitschriften und -Leserdienste begannen allmählich mit ersten Fachbeiträgen oder der Einrichtung von Fachsparten.

Nun, nach etwa 10 Jahren im „TR- und μP-Zeitalter", ist die Lage völlig anders. Die Fülle vor allem der englischsprachigen Fachliteratur ist kaum noch zu bewältigen. Der Interessent kann jetzt aus einer Vielzahl von Lehrbüchern, Datenbüchern, Programmsammlungen, Fachzeitschriften etc. wählen.

Auf den folgenden Seiten ist eine Auswahl der bekanntesten und wichtigsten Veröffentlichungen gesammelt. Abschnitt 3.1 listet *Taschenrechner-Bücher* auf, Abschnitt 3.2 *Mikrocomputer-Bücher.* In beiden Fällen wurde nach Möglichkeit das folgende Schema ausgefüllt:

*Autor(en) oder Herausgeber*
*Titel*
*evtl. Buchreihe*
*Verlag oder Vertrieb*
*Erscheinungsort und -jahr*
*(Seiten, Preis)*

Die *Zeitschriften* in Abschnitt 3.3 sind ebenfalls alphabetisch geordnet. Aufgenommen sind reine Fachzeitschriften wie z.B. die deutsche „Chip" oder die amerikanische „Kilobaud", aber auch allgemeine Elektronik-Zeitschriften mit μP-Teil oder TR- und μP-Aufsätzen wie beispielsweise die „Funkschau". Anders als bei den Büchern ist hier die vollständige Verlagsanschrift angegeben. Bei amerikanischen Zeitschriften ist zum Teil der europäische oder deutsche Auslieferer genannt. Hinter dem Zeitschriftentitel ist in Klammern angegeben, wieviele Ausgaben pro Jahr (pa: *pro anno*) ausgeliefert werden.

## 3.1 Taschenrechner-Bücher

*Alt, H.:*
Angewandte Mathematik, Finanzmathematik, Statistik, Informatik für UPN-Rechner.
Verlag Vieweg, Braunschweig 1979
(162 Seiten, DM 29,80)

*Athen, H. und Bruhn, J.:*
Blitzrechnen mit dem Elektronik-Taschenrechner.
Bertelsmann Ratgeberverlag, München 1975
(94 Seiten, DM 8,–)

*Bacher, M.:*
ARISTO – Electronic-Rechner im Unterricht.
Aristo-Schriftenreihe, Hamburg 1975
(32 Seiten)

*Baier, E.:*
Programmieren von Taschenrechnern SR-56, SR-52.
Verlag Wilhelm Ennsthaler, Steyr 1977
(112 Seiten, DM 9,80)

*Bromm, U.:*
Programmierbare Taschenrechner in Schule und Ausbildung.
Verlag Vieweg, Braunschweig 1979
(200 Seiten, DM 27,–)

*Buckel, F. W.:*
Rechnen mit Stab und Taschenrechner.
Hueber-Holzmann Verlag, München 1975
(200 Seiten, DM 18,–)

*Diepold, P.:*
Taschenrechner-Programme zur Statistik.
Verlag H. Deutsch, Thun 1979
(96 Seiten, DM 12,80)

*Eisberg, R. M.:*
Mathematische Physik für Benutzer programmierbarer Taschenrechner.
Oldenbourg-Verlag, München 1978
(190 Seiten, DM 29,80)

*Fleischhauer, P.:*
Aufgaben lösen und Spiele mit dem Taschenrechner.
Falken-Verlag, Niedernhausen 1977
(120 Seiten, DM 9,80)

*Gloistehn, H. H.:*
Lehr- und Übungsbuch für den SR-56.
Verlag Vieweg, Braunschweig 1977
(140 Seiten, DM 19,80)

*Gloistehn, H. H.:*
Lehr- und Übungsbuch für den TI-57.
Verlag Vieweg, Braunschweig 1978
(112 Seiten, DM 19,80)

*Gloistehn, H. H.:*
Lehr- und Übungsbuch für den TI-58 und TI-59.
Verlag Vieweg, Braunschweig 1978
(150 Seiten, DM 22,80)

*Hansen, K.:*
Rechnen mit dem Electronic-Rechner M 85.
Aristo-Schriftenreihe,
Hamburg 1977
(32 Seiten)

*Henrici, P.:*
Analytische Rechenverfahren für den Taschenrechner HP-25.
Oldenbourg-Verlag,
München 1978
(230 Seiten, DM 29,80)

*Hoyer, K. und Schnell, G.:*
Differentialgleichungen der Elektrotechnik (Lösung mit programmierbarem Taschenrechner).
Verlag Vieweg,
Braunschweig 1978
(120 Seiten, DM 16,80)

*Lisle, Chr. de:*
Elektronik-Rechner in der Management-Praxis.
Sinclair Radionics Ltd.,
Cambridge 1974
(100 Seiten)

*Morgenstern, W.:*
Taschencomputer-Handbuch

*Müller, H. J.:*
Taschenrechner.
Humboldt-Taschenbuch
Nr. 292
München 1976
(128 Seiten, DM 4,80)

*Osterloh, H.:*
Die Wahl des richtigen Taschenrechners.
Bauverlag,
Wiesbaden 1977
(64 Seiten, DM 8,–)

*Sacher, W.:*
Einführung in die Statistik für Benutzer programmierbarer Taschenrechner.
Oldenbourg-Verlag,
München 1977
(126 Seiten, DM 18,80)

*Schärf, J., Schierer, H. und Strecha, R.:*
Die Taschenrechner TI-30, TI-45 und SR-40 in Schule und Praxis.
Oldenbourg-Verlag,
München 1977
(104 Seiten, DM 14,80)

*Schärf, J., Schierer, H. und Strecha, R.:*
Die Taschenrechner SR-50 A und SR-51 A in Schule und Praxis.
Oldenbourg-Verlag,
München 1977
(ca. 130 Seiten, DM 19,80)

*Schärf, J., Schierer, H. und Strecha, R.:*
Der Taschenrechner SR-51 II in Schule und Praxis.
Oldenbourg-Verlag,
München 1977
(136 Seiten, DM 19,80)

*Schärf, J., Schierer, H. und Baron, W.:*
Programmieren mit dem Taschenrechner SR-56.
Oldenbourg-Verlag,
München 1978
(150 Seiten, DM 19,80)

*Schärf, J. und Strecha, R.:*
UPN-Taschenrechner in Schule und Praxis.
Oldenbourg-Verlag,
München 1977
(190 Seiten, DM 23,80)

*Schauer, H. und Barta, G.:*
Programmieren von Tisch- und Taschenrechnern.
Springer-Verlag,
Wien 1978
(200 Seiten)

*Schlossberg/Brockman:*
Spiel und Spaß mit dem Taschenrechner.
Mosaik-Verlag,
München 1976
(160 Seiten)

*Schumny, H.:*
Taschenrechner-Handbuch.
Verlag Vieweg,
Braunschweig 1977
(132 Seiten, DM 18,–)

*Schumny, H.:*
Taschenrechner in der Schulpraxis am Beispiel des schiefen Wurfs, Beiträge zum technischen Unterricht 1.
Verlag Vieweg,
Braunschweig 1977
(24 Seiten, DM 2,50)

*Smith, J. M.:*
Scientific Analysis on the Pocket Calculator.
John Wiley & Sons, 1975
(380 Seiten)

*Texas Instruments:*
Das große Internationale Mathematik über Tasten Buch.
Freising 1977
(168 Seiten, DM 14,80)

*Texas Instruments:*
Entscheidungen rechnerisch ermitteln.
Freising 1978
(170 Seiten, DM 14,80)

*Texas Instruments:*
Mathematik für alle Tage.
Freising 1976
(52 Seiten, DM 4,80)

*Texas Instruments:*
Programmieren von Taschenrechnern.
Freising
(110 Seiten)

*Thießen, P.:*
Lehr- und Übungsbuch für die Rechner HP-29C/HP-19C und HP-67/HP-97.
Verlag Vieweg,
Braunschweig 1979

*Venz, G.:*
Lösung von Differentialgleichungen mit programmierbaren Taschenrechnern.
Oldenbourg-Verlag,
München 1978
(146 Seiten, DM 26,80)

*Wynands, H. und U.:*
Elektronische Taschenrechner in der Schule.
Verlag Vieweg,
Braunschweig 1978
(126 Seiten, DM 19,80)

## 3.2 Mikrocomputer-Bücher

*Ammon, U. v.:*
µP-Lexikon.
Vogel-Verlag,
Würzburg 1978
(56 Seiten, DM 9,–)

*Bender, K., Heinzel, W., Jakob, H., Motsch, W. und Weber, W.:*
Mikrorechner; Struktur und Programmierung.
VDI-Verlag
Berlin 1977
(170 Seiten, DM 44,–)

*Bernstein, H.:*
Hochintegrierte Digitalschaltungen und Mikroprozessoren.
Hofacker-Verlag,
München 1976
(568 Seiten, DM 82,–)

*Bernstein, H.:*
Mikroprozessor Teil 1: Grundlagen, Eigenschaften und Aufbau.
Hofacker-Verlag,
München 1976
(122 Seiten, DM 19,80)

*Bernstein, H.:*
Mikroprozessor Teil 2.
Hofacker-Verlag,
München
(120 Seiten, DM 19,80)

*Bernstein, H.:*
TTL-Mikroprozessor-System.
Frech-Verlag,
Stuttgart
(103 Seiten, DM 8,–)

*Birchel, R.:*
Vom Gatter zum Mikroprozessor. Sonderdruck Elektronik-Entwicklung.
Verlag f. Technik und Wirtschaft,
Wiesbaden 1978
(146 Seiten, DM 10,80)

*Blomeyer-Bartenstein, H.-P.:*
Mikrocomputer Technik.
Hofacker-Verlag,
München
(240 Seiten, DM 29,80)

*Blomeyer-Bartenstein, H.-P.:*
Mikroprozessoren und Mikrocomputer.
Siemens AG,
München 1977
(64 Seiten)

*Burghardt, K.:*
Der Mikroprozessor vom Bauteil zur Anwendung.
Alfred Neye,
Quickborn
(135 Seiten, DM 42,60)

*Butterfield, J. et. al:*
The First Book of KIM.
F. J. Butterfield,
Feltron Elektronik 1977
(176 Seiten, DM 20,–)

*Diehl, W.:*
Mikroprozessoren und Mikrocomputer. Kamprath-Reihe kurz und bündig,
Vogel-Verlag,
Würzburg 1977
(200 Seiten, DM 30,–)

*Dirks, Chr. und Kirnn, H.:*
Mikrocomputer.
Verlag Berliner Union,
Stuttgart 1977
(208 Seiten, DM 49,–)

*Dittmar, E.:*
Mikroprozessor-Einsatz in der Automatisierung.
Vogel-Verlag,
Würzburg 1979
(144 Seiten, DM 25,–)

*Görke, W.:*
Mikrorechner.
BI Wissenschaftsverlag,
Reihe Informatik Nr. 26,
Zürich 1978
(226 Seiten, DM 26,80)

*Gräf, R. und Kammerer, J.:*
Elektronik IV C – Mikrocomputer-Lehrbuch.
Pflaum-Verlag,
München 1979
(360 Seiten, DM 48,–)

*Gräf, R. und Kammerer, J.:*
Elektronik VI C – Mikrocomputer-Arbeitsblätter.
Pflaum-Verlag,
München 1979
(150 Blätter, DM 12,–)

*Hatzenbichler, J.:*
Mikrocomputer Programmierbeispiele für 2650.
Hofacker-Verlag,
München
(120 Seiten, DM 19,80)

*Heubach, F.:*
Valvo-Mikroprozessoren, Wörterbuch.
Valvo-Bauelemente,
Hamburg 1976
(56 Seiten)

*Hilberg, W. und Piloty, R.:*
Mikroprozessoren und ihre Anwendungen.
Oldenbourg-Verlag,
München 1977
(283 Seiten, DM 58,–)

*Hobby-Computer,*
Sonderheft der ELO, Funkschau, Elektronik.
Franzis-Verlag,
München 1978
(136 Seiten, DM 19,–)

*ICS:*
Lexikon der Mikroelektronik
IWT-Verlag,
München 1978
(784 Seiten, DM 137,–)

*Klein, R.-D.:*
Mikrocomputersysteme.
Franzis-Verlag,
München 1978
(160 Seiten, DM 29,–)

*Kobitzsch, W.:*
Mikroprozessoren – Aufbau und Wirkungsweise, Teil 1: Grundlagen.
Oldenbourg-Verlag,
München 1977
(160 Seiten, DM 38,–)

*Kobitzsch, W.:*
Mikroprozessoren – Aufbau und Wirkungsweise, Teil II: Anwendungen und Beispiele.
Oldenbourg-Verlag,
München 1978
(200 Seiten, DM 40,–)

*Lesea, A. und Zaks, R.:*
(Mikroprocessor Interfacing Techniques) Microprozessor-Interface-Techniken.
SYBEX Europe
Paris 1978
(414 Seiten, DM 39,–)

*Lorenz, C.:*
Hobby Computer Handbuch.
Hofacker Verlag,
München
(450 Seiten, DM 29,80)

*Lorenz, C.:*
Lexikon und Wörterbuch für Elektronik und Mikroprozessortechnik.
Hofacker-Verlag,
München 1978
(250 Seiten, DM 29,80)

*Lorenz, C.:*
Mikrocomputer Datenbuch.
Hofacker-Verlag,
München 1978
(DM 49,–)

*Lorenz, C.:*
Mikroprozessor Software Handbuch.
Hofacker-Verlag,
München
(200 Seiten, DM 29,80)

*Lorenz, C.:*
57 Programme in BASIC.
Hofacker-Verlag,
München 1978
(DM 39,–)

*Martin, P.:*
Microcomputer Design.
Optima Supply Comp.
Allensbach 1977
(400 Seiten, DM 46,–)

*Mikrocomputer-Bauteilekatalog.*
Feltron Elektronik,
Postfach 1169
5210 Troisdorf
(Loseblattsammlung, kostenlos)

*Mikrocomputer-Software.*
Feltron Elektronik,
Postfach 1169
5210 Troisdorf
(Loseblattsammlung, Grundpreis DM 43,–)

*Mikroprozessor-Guide.*
Spoerle Electronic
Otto-Hahn-Straße 13
6072 Dreieich
(32 Seiten, kostenlos)

*Osborne, A.:*
Einführung in die Mikrocomputer-Technik.
Te-wi-Verlag,
München 1978
(400 Seiten, DM 66,–)

*Osborne, A.:*
An Introduction to Microcomputers – Volume 2 – Some Real Microprocessors.
A. Osborne & Ass.
Te-Wi-Verlag,
München
(1200 Seiten, DM 86,–)

*Osborne, A.:*
An Introduction to Microcomputers – Volume 3 – Some Real Support Devices.
A. Osborne & Ass.
Te-wi-Verlag,
München
(700 Seiten, DM 56,–)

*Osborne, A.:*
Mikrocomputer-Grundwissen.
Te-wi-Verlag,
München 1978
(300 Seiten, DM 36,–)

*Pelka, H.:*
Praxis mit Mikroprozessoren.
Franzis-Verlag,
München 1977
(207 Seiten, DM 19,80)

*Pelka, H.:*
Was ist ein Mikroprozessor?
RPB electronic taschenbücher Nr. 82
Franzis-Verlag,
München 1979
(128 Seiten, DM 7,80)

*Sacht, H.-J.:*
Mikroprozessoren.
Humboldt-Taschenbuch Nr. 338,
München 1979
(ca. 120 Seiten, DM 4,80)

*Schlossberg, E., Brockman, J. und Horton, L.:*
The Home Computer Handbook.
Bantam Book,
New York 1978
(250 Seiten, $ 2.95)

*Schneider, W.:*
BASIC; Einführung für Techniker.
Verlag Vieweg,
Braunschweig 1979
(126 Seiten, DM 19,80)

*Schumny, H.:*
Mikroprozessoren, Beiträge zum technischen Unterricht 3.
Verlag Vieweg,
Braunschweig 1978
(24 Seiten, DM 2,50)

*Söll, W. und Kirchner, J.-H.:*
Digitale Speicher
Kamprath-Reihe kurz und bündig.
Vogel-Verlag,
Würzburg
(220 Seiten, DM 38,–)

*SYBEX:*
Microprocessor Lexicon.
SYBEX Europe,
Paris 1978
(110 Seiten, DM 8,–)

*Texas Instruments:*
Application Handbook 1,
TMS 9900 Microprocessor Family.
Freising 1977
(290 Seiten, DM 14,80)

*Texas Instruments:*
Applikationsbuch. Band II.
Freising 1977
(DM 17,90)

*Texas Instruments:*
Designing with Microprocessors; Lesson Summaries.
Dallas 1975
(368 Seiten)

*Texas Instruments:*
Einführung in die Mikroprozessor-Technik.
Freising
(370 Seiten, DM 64,–)

*Texas Instruments:*
TM 990/100 M Mikrocomputer-Anwender-Handbuch.
Freising 1977
(DM 17,50)

*Texas Instruments:*
9900 Family System Design and Data Book.
Freising 1979
(1064 Seiten, DM 28,80)

*Tireford, H.:*
Vom Computer zum Mikroprozessor.
Motorola Inc.,
Wiesbaden 1975
(58 Seiten, ca. DM 5,–)

*Veronis, A.:*
Microprocessors: Design & Applications.
Reston Publishing,
USA 1978
(260 Seiten, $ 22.75)

*Vogel-Verlag:*
Mikroprozessor-Praxis.
Sonderband der „elektronikpraxis".
Würzburg 1978
(170 Seiten, DM 19,80)

*Zaks, R.:*
An Introduction to Personal and Business Computing.
SYBEX Europe
Paris 1978
(246 Seiten, DM 19,–)

*Zaks, R.:*
Microcomputer Programming: 6502.
SYBEX Europe
Paris 1978
(250 Seiten)

*Zaks, R.:*
Microprocessors from Chips to Systems.
SYBEX Europe
Paris 1978
(420 Seiten, DM 26,–)

*Zaks, R.:*
Programming the 6502.
SYBEX Europe
Paris
(305 Seiten, DM 26,–)

## 3.3 Zeitschriften

**bit, büro + informationstechnik**
(12 Ausgaben pa)
bit-Verlag
Hans W. Ulbricht GmbH
Sofienstraße 28
7570 Baden-Baden

**Byte**
(12 Ausgaben pa)
Byte Publications Inc.
70 Main Street
Peterborough, NH 03458, USA

**Chip**
Zeitschrift für Mikrocomputer-Technik
(12 Ausgaben pa)
Vogel-Verlag
Postfach 6740
8700 Würzburg 1

**Computer Design**
The Magazine of Digital Electronics
(12 Ausgaben pa)
11 Goldsmith Street
Littleton, MA 01460, USA
und
KLM Publications Handling Dept.
P. O. Box 75 200
1117 Zt Schiphol, Holland

**Computerworld**
(52 Ausgaben pa)
Computerworld Inc.
797 Washington St.
Newton, Mass. 62160, USA

**Creative Computing**
(6 Ausgaben pa)
Creative Computing,
P. O. Box 789-M,
Morristown, NJ 07960, USA

**Datamation**
(12 Ausgaben pa)
Technical Publishing Co.
1301 S. Grove Ave.
Barrington, Ill. 60010, USA

**Der Elektroniker**
(12 Ausgaben pa)
AT-Fachverlag,
Postfach 50 0180
7000 Stuttgart 50

**Die Computer Zeitung**
(24 Ausgaben pa)
Konradin-Verlag,
Robert Kohlhammer GmbH
Postfach 625
7000 Stuttgart 1

**Dr. Dobb's Journal**
of Computer Calisthenics & Orthodontis
(12 Ausgaben pa)
People's Computer Company
P. O. Box 310
Menlo Park, Cal. 94025, USA

**EDN/Cahners**
(10 Ausgaben pa)
221 Columbus Avenue
Boston, MA 02116, USA
und
KLM Publications Handling Dept.
P. O. Box 75 200
1117 Zt Schiphol, Holland

**Elcomp**
Fachzeitschrift für Mikrocomputer
(10 Ausgaben pa)
Hofacker Verlag,
Tegernseerstraße 18
8150 Holzkirchen

**Electronic Design**
(23 Ausgaben pa)
Hayden Publishing Comp.
50 Essex Street
Rochelle Park, NJ 07662, USA

**Electronic Engineering**
(16 Ausgaben pa)
Morgan-Grampian Limited,
30 Calderwood Street
London SE 186 QH, England

**Electronics**
(26 Ausgaben pa)
Mc Graw-Hill Inc.
1221 Avenue of the Americas
New York, NY 10020, USA

**Elektronik**
Fachzeitschrift für angewandte Elektronik und Datentechnik
(26 Ausgaben pa)
Franzis-Verlag,
Postfach 37 0120
8000 München 37

**Elektronik Praxis**
(12 Ausgaben pa)
Vogel-Verlag,
Postfach 6740
8700 Würzburg 1

**Elektronische Rechenanlagen**
mit Computer-Praxis
(6 Ausgaben pa)
Oldenbourg-Verlag,
Rosenheimer Straße 145
8000 München 80

**Elrad**
Magazin für Elektronik
(12 Ausgaben pa)
Verlag Heinz Heise,
Postfach 2746,
3000 Hannover 1

**Euromicro Journal**
International Journal of Microprocessing and Microprogramming.
(6 Ausgaben pa).
North-Holland Publishing Comp.
P. O. Box 211,
1000 AE Amsterdam, Holland

**FELTRON:**
**Microcomputer-Bauteile**
(Loseblattsammlung, kostenlos)
FELTRON-Elektronik
Postfach 1169
5210 Troisdorf

**FELTRON:**
**Microcomputer-Information**
(Loseblattsammlung)
FELTRON-Elektronik
Postfach 1169
5210 Troisdorf

**FELTRON:**
**Microcomputer-Software**
(Loseblattsammlung)
FELTRON-Elektronik
Postfach 1169
5210 Troisdorf

**Funkschau**
(26 Ausgaben pa)
Franzis-Verlag
Postfach 37 0120
8000 München 37

**Interface Age**
(12 Ausgaben pa)
McPheters, Wolfe & Jones
6615 Sunset Blvd.
Hollywood, Cal. 90028, USA

**Kilobaud Microcomputing**
(12 Ausgaben pa)
Peterborough, NH 03458, USA
in Deutschland:
Micro-Shop Bodensee
Marktstraße 3
7778 Markdorf

**Markt & Technik**
Wochenzeitung für Elektronik
(52 Ausgaben pa)
Verlag Markt & Technik,
Kuglmüllerstraße 25
8000 München 19

**Micro Computer Journal Joce & N**
(12 Ausgaben pa geplant)
Computer Club Europe
Postfach 11 0868
6100 Darmstadt

**MICRO EXTRA**
Zeitschrift für Microcomputertechnik
(6 Ausgaben pa)
FELTRON Elektronik
Auf dem Schellerod 22
5210 Troisdorf

**65XX Micro Mag**
Computing, Software, Hobby
(6 Ausgaben pa)
Dipl.-Volkswirt R. Löhr,
Hansdorfer Straße 4
2070 Ahrensburg

VIEWEG

## Zum Thema:
## Taschenrechner + Mikrocomputer

**Taschenrechner + Mikrocomputer Jahrbuch 1981**
Anwendungsbereiche – Produktübersichten – Programmierung – Entwicklungstendenzen – Tabellen – Adressen.
Herausgegeben von Harald Schumny. Mit zahlr. Abb. 1980. ca. 270 S., 18,5 X 24 cm. Kart. ca. 22,80 DM.

**Programmieren von Heimcomputern**
Wolfgang Schneider
**Einführung in BASIC**
Mit zahlr. Beisp. u. 10 völlst. Programmen. 1979. Ca. 130 S. DIN C 5 (Programmieren von Heimcomputern, Bd. 1). Kart. ca. 20,– DM.

Harald Schumny
**Taschenrechner Handbuch**
Naturwissenschaften/Technik. Mit 16 Abb. 2., durchges. Aufl. 1978. 132 S. Kart. 19,80 DM.

Allgemeine Anleitung zur Auswahl und Gebrauch elektronischer Taschenrechner.

# Sachwortverzeichnis


- Abakus 1
Absolute Adresse 86, 127
Active Element Group (AEG) 62
ADC (Analog Digital Converter) 59, 96
Address Latch Enable 93
Adreßbus 70 f., 87, 121
Adreß-Decodierung 86 f., 89 ff.
Adressierung 49, 86 ff.
Adressierung dynamischer Speicher 94
Adreßraum 87 ff., 123
Adreßteil 87, 149
Adreßvolumen 123
Adreßzähler 139
ADV (Automatische Datenverarbeitung) 58
Ähnlichkeitsdecodierung 165
Äquivalenzprinzip 25
AIM 172
Akkumulator 44, 45, 50, 70, 122
Alarmeingang 95
Alarmmeldung 96
Algebraische Logik 44
Algebraische Notation 36, 50
Algebraisches Operations-System (AOS) 44 ff., 46
Algorithmisches Denken 43
Algorithmus 49, 104
ALPHA-Schreibverfahren 156
Altgrad 38
ALU (Arithmetic Logic Unit) 56, 70, 123, 141 f.
Analogdatenverarbeitung 58
Analog Digital Converter (ADC) 59, 96
Analog-Digital-Wandler 59, 96
Analogrechner 59
Anbindung 73
Anfangswert 113
Anforderungskriterien an Schulrechner 46 f.
ANSI 108
Answer File 66
Anweisung 109
Anweisungsnummer 108
Anwendungsbereiche (für Rechner) 58
Anwendungssoftware 62
Anzeige 35 f.
Anzeigencodierer 44, 45
AOS (Algebraisches Operations-System) 44 ff., 46
Approximation 21
APU 146 f.
Arbeitsgeschwindigkeiten (μP) 143
Arbeitsregister 121
Arbeitsspeicher 56
Arbeitsspeichereinheit (ASE) 49, 50
Architektur (μP) 138
Arcus-Tangens 38
Arithmetic Logic Unit (ALU) 56, 123, 141 f.
Arithmetic Processing Unit (APU) 146 f.
Arithmetik-Prozessor 146
Arithmetische Ausdrücke 111
Arithmetische Verknüpfungen 70, 111
Arithmetisch-Logische Einheit 70, 123, 141 f.
ASCII-Code 81, 150
ASCII-Tastatur 68, 150
Assemblerlisting 135
Assembler-Niveau 34
Assemblersprache 72, 92
Auffang-Flipflop 141
Aufzeichnungsverfahren 155 ff.
Ausgabe (Output) 55
Ausgabeanweisungen 115
Ausgabekanal 122
Ausgangsfächerung 177
Ausgangsregister 130
Außenwelt 79
Austauschbarkeit 134
Auswahlkriterien (TR) 32, 35, 44 ff.
Automatische Dateneingabe 56
Automatische Datenverarbeitung 55

- Bandspeicher 182
BASIC 108 ff.
Basis 38
Basis-Umwandung von Zahlen 21 ff.
Baud (Bd) 127, 156
Baudrate 127
BDI-Technologie 178
Bedingter Sprung 49, 140
Bedingte Verzweigung 140
Befehl 108
Befehlsadresse 86
Befehlsausführungszeit 69, 146
Befehlscode 149
Befehlsformat 150
Befehlsregister 122 f.
Befehlsstruktur 87
Befehlszähler 70, 86, 124
Befehlszykluszeiten 144
Betriebssystem 59, 71, 121, 135
Bildladung 175
Binär-Code 21, 72
Binärzustände 21
Bipolare μP 138 ff.
Bipolare Speicher-Technologien 177 ff.
Bipolar-PROM 172
Bipolartechnik 166
Bissaker 5
Bitorientiertes Steuerungsverfahren 161
Bit-Packungsdichte 157
Bitscheiben-Prozessoren 138 ff.
Bit-Slice-Prozessor 71, 138 ff.
Black Box 40, 125
Blasenspeicher 182
Blockende-Interrupt 96
Blockprüfzeichen 161
Blockzwischenraum 163
Bogenmaß 38
Bootstrap 75
Branch 139
Breakpoint 74, 77
Burst 132
Bus 121
Busanordnungen 79, 82 ff.
Busbreite 71
Bus Extender 85
Business Computer 59, 67
Buslänge (IEC-Bus) 85
Bus-Typen 82
Byte 122

- CALL 101
Carry 139
Carry Look Ahead 143, 144
Cathode Ray Tube (CRT) 133
CCD 166, 170, 181 ff.
CCD-Halbleiterspeicher 181 ff.
CCITT 81
CDI-Technologie 178
Central Processing Unit (CPU) 50, 55 f.
Charge Coupled Device (CCD) 166
Chip 138
Chip Select 88, 89, 146, 167
Clear Memory 37
Clock (CLK) 146
Closed Loop 59
CMOS 179
Code-Wandler 172, 173
Column 172
Compiler 150
Computerband 158
Computerstruktur 49
Conditional Branch 140
Conditions 140
Continue 139, 140
Controller 67, 84, 133, 144, 159
Control Unit 56
Cotangens 38
CPU (Central Processing Unit) 50, 55 f.
CRC-Prüfzeichen 165
CRT 133
CRT-Display 17
CTD 170
Current Loop 84
Cursor 73

- DAC (Digital Analog Converter) 59, 96
Darlington-Stufe 136
Darstellung von Programmabläufen 102 ff.
Datei 63
Daten 56
Datenadreßregister 123
Datenbank 58
Datenblock 162
Datenbreite 122
Datenbus 70 f., 87, 121
Datendarstellung (Kassette) 162
Dateneingabe 56
Datenendeinrichtung 80
Datenfeld 161
Datenfernübertragung 56
Datenfluß 56
Datenleitungen 84
Datenmanipulation 16
Daten-Multiplexer 142
Datenrate 59
Datenregister 17, 130
Datenrichtungsregister 130
Datensicherung 159, 160 ff., 164 f.
Datenspeicherung 160 ff.
Datenterminal 135
Datenträger 155 ff.
Datentransfer 57
Datenübertragung 160
Datenübertragungsblock 161
Datenübertragungseinrichtung 80
Datenverarbeitungsanlage 55
DDV (Dialog-Datenverarbeitung) 59
Debugging 71
Decoder 92
Decodierer 92, 123
Decodierung 56, 87
Dekadischer Logarithmus 38
Dekrementierung 16, 70, 97
Destination 139
Dezibel (dB) 132
Dezimalschreibweise 109
Dezimalzahlen 109
Dialog-Datenverarbeitung (DDV) 59
Dialog Mensch/Maschine 59, 62
Dialog mit Heimcomputer 108
Dialogrechner 62
Dialogsprache 108 ff.
Dienstprogramm 73
Digital Analog Converter (DAC) 59, 96
Digitale Aufzeichnungsverfahren 158 f.
Digital-Ohmmeter 16
Digitalrechner 59
Digitron-Anzeige 36
DIL 137
DIM-Anweisung 110
DIN 84, 158
Direct Memory Access (DMA) 57, 96
Direkte Adressierung 49
Direkter Speicherzugriff 57
Direktzugriff 63
Disassembler 72, 135
Diskette 172
Display 17, 65, 133
Distraktoren 7
Distraktorenanalyse 7, 12
Distributor 130
DMA (Direct Memory Access) 57, 96
DMA-Controller 96
DMA-Interrupt 96

DMA-Kanal 57
Don't-Care-Bit 90
Drain 173
Druckwiderstand (TR-Tastatur) 35
Dual-In-Line (DIL) 137
Dualzahlen 21
Dump 77
Durchlaufzeiten 141
Durchschnitt 10
Dynamische Speicher 94, 166, 168 f.
3M-Kassette 158

- EAROM 166, 169, 171
E/A-Schnittstellen 86
ECC 159, 165
Echtzeit 59
Echtzeitanalyse 74, 78
Eckert 1
ECL-Sequenzer 141
ECMA 108, 158
Editieren 34
Edit Mode 74
Editor 73
EDV-System 55
EIA 81, 127
Eimerkettenschaltung 181
Ein-Ausgabe-Interrupt 96
Ein-Ausgabe-System 92 f.
Einchip-μP 69, 71, 138
Einerkomlement 7
Einfache Variablen 110
Eingabe (Input) 55 f.
Eingabeanweisungen 114
Eingabe/Ausgabe-Kanäle 70
Eingabekanal 122
Einkommensteuer 25 ff.
Einscheiben-Prozessor 138
Einzelschrittverfahren 74, 135
Elektronische Datenverarbeitung (EDV) 55
Emulation 72
Emulator 74, 137, 144
Enable 93, 139, 180
END-Anweisung 114
Endwert 114
ENIAC 1
Enter Exponent 36
Enter-Taste 28, 36
Entwicklungssystem 69 ff., 72
Entwicklungstendenzen (TR) 32, 34
EPROM 137, 166, 169, 171
Error Correcting Code (ECC) 159
ESC (Escape) 150
ETX (End of Text) 109
Eulersche Zahl 38
Euro Board Computer 136
Euro-Diskette 159
Europakarte 135
Evolution 61
Execute 139
Exemplifizieren 43
Exorciser 145
Exponentielle Zahlendarstellung 36
Externe Speicher 5, 6

- False 140
FAMOS 169, 172
Fan Out 177
Fehler 41
Fehlerbeseitigung 71
Fehlererkennung 164, 165
Fehlerfortpflanzung 41
Fehlerkorrektur 165
Fehlermeldung 37 f., 109
Fehlerquittung 161
Feld 110
Feldeffekttransistor 168
Fernschreibgeschwindigkeit 156
Festkomma-Arithmetik 146
Festverdrahtete Logik 60
Festwertspeicher (ROM) 17, 38, 71, 125, 166, 169
Festwertspeicherzelle 180
FET 168
File 56
File Header 66
Firmware 137
Flächenspeicherdichte 158
Flag 70, 147
Flagregister 124
Flexible Magnetplatte 158, 163 f.
Fließkommadarstellung 36
Flipflop 92, 170
Floppy Disk 158, 163 f.
Floppy-Speicher 182
Flüchtige Speicher 167
Flüssigkristall-Anzeige (LCD) 34, 36
Flußdiagramm 49
FM 156
Format 66, 115 f., 161 ff.
Format, Flexible Magnetplatte 163 f.
Format, Magnetbandkassette 161 f.
Formatieren 34
Formatierung 151, 158, 162
Formatter 159
FOR ... NEXT 113
Fotostrom 175
Free Running Mode 131
Freilauf 131
Frequency Shift Keying (FSK) 156
Frequenzcodierung 156
Frequenzmodulation (FM) 156
Frequenztastung 156
Frequenzumtastung 156
FSK 156
Fünferkomplement 7
Funktionen 40
Funktionsgruppen (μC) 123
Funktionstasten 38
Fuse 172
Fusible Link 172
Φ (Phase) 146

- Ganze Zahlen 109
Gate 173, 181
Gatterfunktion 34, 172
GCR 158
Gegentakt-Endstufe 180
Generalisieren 43
General Purpose Computer (GPC) 2, 76
Geschlossene Regelschleife 59
Gleitkomma-Arithmetik 146
Glitch 78
GOTO 112
GPC (General Purpose Computer) 2, 67
Grafische Ablaufpläne 57, 102 ff.
Grow Back 175
Grundzahl 38

- H (High) 21, 87
Halbleitereffekt 61
Halbleiterlesespeicher 64
Halbleiterspeicher 166 ff.
Handshake 70, 71, 84, 147
Handshake-Leitungen 130
Handshake-Leitungen (IEC-Bus) 84
Hardware 6, 49, 55 f., 60
Hardware/Software-Integration 73
Hauptbereiche der ADV 58
Hauptbestandteile eines TR 44 f.
Hauptprogramm 149
Hauptspeicher 56, 123
HDLC 161
Head 160
Heimcomputer 2, 59, 67 ff., 108
Hexadezimal-Code 21
Hexadezimalzahl 72
Hexcode 127
Hierarchie 36, 44, 46
High 87
High-Pegel 146
Hilfsspeicher 56
History 78
Hobby-Computer 2, 59
Hobby-Computing 59, 67 ff.
Hörfrequenzbereich 155
Hollerith 1
Home Computer 2, 59, 67 ff.
HP-Interface Bus 84
Hybridrechner 60
Hypertape 156

- IC (Integrated Circuit) 60, 63, 171
IEC 84
IEC-Bus 84 f., 136
IEEE 84
IEEE 488-Bus 84 f.
$I^2L$ 166
Image Charge 175
Implementierung 135
Impulserzeugung 130 ff.
Impulsmuster 128
In Circuit Emulation 72, 78
Incrementer 140
Index 8, 106
Index-Spur (Floppy) 163
Indirekte Adressierung 16, 49
Individueller Dialogrechner 62
Indizes 8
Indizierte Varablen 110
Influenz 175
Informatik-Unterricht 121
Initialisierung 74, 131
Injected Electrons 173
Injektion 173
Inkrementierung 16, 70, 97
Innovation 6
Input (Eingabe) 55 f.
INPUT-Anweisung 115
Input Unit 57
Instruktion 56, 122
Instruktionseingabe 121 f.
Instruktionssatz 138
Integer 21
INTEGER-Zahlen 109
Integrierter Baustein 60
Integrierter Bus 82
Integrierter Schaltkreis 63
Interactive Computer 59
Interaktive Verarbeitung 62
Interface 51, 79 ff., 134
Interface-Adapter 130
Interfacebus 84
Interpreter 150
Interrupt 70, 71, 86, 95 ff.
Interruptmaske 96
Interruptprogramm 88
Interrupt Request (IRQ) 95
Interrupt Service Routine 88, 95
Intervallschachtelung 41
Inverter 34
I/O (Input/Output) 70, 71, 136, 144
I/O-Bus 146
IRQ 95
ISO 81, 158
Isoplanar-Technik 179

- Ja-Nein-Entscheidung 72, 105, 106
Jump 139

- K = 1024 135
Kanal 122
Kansas-City-Schreibverfahren 157
Kaskadierung 71, 138
Kassetten 158
Kassettenrecorder 155
KC-Standard 157
Kernspeicher 56, 166, 182
Keyboard 133
KIM-Schreibverfahren 156
Kit 71
Klammerautomatik 44, 46
Klammerhierarchie 46
Klassifizierung von Speichern 166
Koeffizient 8
Kommentare 117 f.
Kommerzielle DV 58
Kompatibles System 134
Konstanten 109
Konstantenautomatik 37
Kontrolleitungen (IEC-Bus) 84
Kontrollspeicher 138, 139
Kopplungselektronik 60
Kurzwegtechnik 45

- L (Low) 21, 87
Label-Technik 17
Ladungsgespeicherte Einheiten 166
Ladungspakete 181
Längsparität 159, 164
Längsprüfung 164
Large Scale Integration (LSI) 34, 62
LAST X-Register 28
Latch 93, 130
Latching 130
Latchregister 141
Laufbereich 113
Laufvariable 113
Lawinen-Durchbruch 172
LCD (Liquid Crystal Display) 34
Lehrsystem (μC) 126 ff.
Leibniz 5
Leistungsfähigkeitsprinzip 25
Leitwerk 56
Lerncomputer 121 ff.
Lernsystem (μC) 121 ff., 126 ff.
Lesesignal 147
LET-Anweisung 112
Leuchtdioden-Anzeige 36

Lineare Anweisungsfolge 57
Lineare Programmierung 49
Linking 73
Link-Prozeß 73
Liquid Crystal Display (LCD) 34
Liste 110
Listener 84
Listing 17, 76, 135
Location 90
Logarithmus 38
Logic High 140
Logikanalysator 72
Logik-Bausteine 60
Logische Entscheidung 57
Logische Fehler 73
Logischer Vergleich 16
Logische Verknüpfungen 70
Logisch 0, 1 140
Low 87
Low-Pegel 146
Low Power Schottky 177
LSI (Large Scale Integration) 34, 62, 63
LSI-Chip 34

- Magnetbandkassette 155, 158, 161 f.
Magnetische Datenträger 155 ff.
Magnetkarte 8, 17, 30
Magnetplatten 158
Main Memory 57
Makro 73
Makro-Assembler 73
Makrobefehl 6, 144, 145
Makrosequenzer 145
Management-Leitungen (IEC-Bus) 84
Manuelle Dateneingabe 56
Map 77
Mapping 78
Mark 81
Maschinen-Code 73
Maschinenfehlerinterrupt 96
Maschinenprogramm 86
Maschinensprache 72
Maschinen-Zykluszeit 69
Masken 171
Maskenbit 96
Masken-ROM 169
Maskierung 96
Massenspeicher 58, 74, 136, 155 ff., 181
Master PROM 171
Master Reset 75
Master-Slave-Prinzip 170
Mathematik-Assembler 149 f.
Matrix 110
Mauchly 1
Maximum Likelihood Detection 165
MC (Mikrocomputer) 134
µC in der Schulpraxis 48 ff.
µC-Lernsystem 121 ff., 126 ff.
Mehrfachbelegung von Tasten 35
Mehrfachübertragung 164
Mehrzweck-Computer 2, 67
Memory 37, 55 f.
Memory-Maped-E/A 92, 93, 147
Memory Mapping 78
Meßwandler 172, 173
MFM 158
Mikrocomputer 58
Mikrocomputer-System 58
Mikrocomputer-Trainings-Set 121
Mikrominiaturisierung 62
Mikroprogramm 138
Mikroprogrammspeicher 139
Mikroprogrammwerk 71, 124
Mikroprozessor 58, 62, 69 ff.
Mikroprozessor-Kit 71
Mikroprozessor-Praktikum 121
Mikroprozessortypen 70 f.
Mikrosequenzer 145
Minicomputer 1 f., 69, 86
Mini-Floppy 159
Minimal BASIC 108
Mittelwert 8
Mnemonische Darstellung 135
Mnemonischer Ausdruck 34, 135
Mnemonischer Code 72
MNOS 175
Modem 74
Modemverbund 74
Monitor 71, 121
Monitorprogramm 121, 125
Monolithischer Schaltkreis 5, 6
Monotonie 41
Morland 5
MOS 172
MOSFET 168, 172
MOS-PROM 172 f.
MOS-Technologie 166
µP-Ausbildung 129
µP-Entwicklungssystem 72
µP-Kit 71
MSI (Medium Scale Integration) 63
Multiple-Choice-Test 7 ff.
Multiplex 93, 142
Multiplikationstabelle 172, 173
Musikrecorder 157
MUX 142

- Nanosekunde 5, 6
Negative Logik 140
Nesting 96, 100 f.
Netzausfallinterrupt 96
Neugrad 38
Nichtflüchtige Speicher 167
NMI 97
Non Maskable Interrupt (NMI) 97
Normschnittstellen 74
Notation 36, 50
NRZI 159
Null-Katastrophe 39
Nur-Lese-Speicher (ROM) 38
19-Zoll-Gehäuse 135

- Objektive Fehler 41
Offener Kollektor 94, 147, 180
One Shot 133
Opcode 149
Open Collector 94, 147, 180
Open-Drain-Leitung 147
Operand 149
Operandenadresse 87
Operation 149
Operationscode 149
Operationsteil 87, 149
Operator 40
Optokopplung 136
Output (Ausgabe) 55 f.
Output Unit 57
Overflow 35
Overheadfolien 135

- Parallele Verarbeitung 59
Parallelport 142
Parallelschnittstelle 74, 82
Parität 74, 159, 164
Paritätsbit 159
Paritätsprüfung 164
Paritätsverletzungsinterrupt 96
Partridge 5
Pascal 5 f.
PC (Program Counter) 97, 124
PCM 155
PDV (Prozeßdatenverarbeitung) 58, 95
PE 159
Pegel 132
Peripheral Control Register 130
Peripheral Interface Adapter (PIA) 130
Peripherie 51, 55 f., 79, 134
Peripheriebus 82, 84
Personal Computer 2, 59, 61 ff., 67 ff.
PET (Personal Electronic Transactor) 68
PET-Schreibverfahren 156 f.
Phase-Encoding (PE) 159
Philips-Kassette 158
PIA 130
Pin 81, 130
Pipelineregister 140, 144
Pipelining 140, 144
Plattenspeicher 182
PMOS 179
POP 98
Port 130, 142
Positionsanzeiger 73
Positive Logik 140
Potentialtöpfe 181
Primärprogramm 102
PRINT-Anweisung 115, 117
Priorität 100
Prioritäts-Netzwerk 100
Problemanalyse 102
Program Counter (PC) 86, 97, 124
Programm 55
Programmablaufplan 8, 10, 57, 102 ff.
Programmausführungszeit 69, 148
Programmbeendungsanweisungen 114
Programmbefehle 56, 122
Programmierbarer Festwertspeicher (PROM) 71, 171 ff.
Programmschalter 121
Programmschleife 113, 124
Programmspeicher 17
Programmspeichereinheit (PSE) 50
Programmspeicherstelle (PSS) 21
Programmsprung 70
Programmunterbrechung 70, 71
Programmverzweigung 57
Programmverzweigungsanweisung 113
Programmzähler 70, 97, 124
PROM 71, 169, 171 ff.
PROM-Typen 172 ff.
Prototyp 136
Prototypenherstellung (µC) 73
Prozentautomatik 38
Prozeßdatenverarbeitung (PDV) 58, 95
Prozeßhardware 136
Prozessor 55 f.
Prozessorstatuswort 70
Prozeßrechner 67, 86
Prüfzeichen 159
PSS (Programmspeicherstelle) 21
PULL 98
Pulscodemodulation (PCM) 155
PUSH 98

- Quelle 139
Quellenprogramm 102
Querparität 159, 164
Querprüfung 164

- Rack 135
RAM (Random Access Memory) 38, 63, 70, 123, 166, 167 f.
Randbedingung 102
Random Access Memory (RAM) 38, 70, 123, 166, 167 f.
READ 147
READ-DATA-Anweisung 114
Read Only Memory (ROM) 38, 64, 71, 125, 166, 169
READY 122, 147
REAL-Zahlen 109
Recall 30, 37
Rechenablaufplan 40
Rechenelemente 59
Rechengenauigkeit (TR) 39
Rechenmaschine 1, 5 f.
Rechenregister-Stapel (Stack) 29, 46
Rechenschieber 6
Rechenuhr 5
Rechenwerk 44, 45, 55, 121
Rechner 69
Rechnerbaustein 45
Rechnerklassen 2, 58 f.
Rechnerprogramm 49
Rechnerstruktur 49
Rechteckfehler 164
Rechtsschieben 142
Recording Modes 158 f.
Redundanz 164
Refresh 94, 167
Register 44, 45, 70
Registersatz 142
Relative Adressierung 49
Reliabilität (Reliability) 8
Reliabilitätskoeffizient 8, 10
REM-Anweisung 118
Rentenformel 118
REPROM 169, 171
Reset 8, 122
Restart (RST) 98, 99
Return (RET) 98
Richtlinien für TR im Unterricht 33
Richtungstaktschrift 159
ROM (Read Only Memory) 38, 64, 70, 125, 166, 169
Rotate 142
Routine 149
Row 172
RS 232 C 81, 127
RST (Restart) 99
Rücksprung 112
Rundungsautomatik 39

- Saldierender Speicher 37
Sandwichbauweise 136
SBC (Single Board Computer) 136
SBC-Technologie 178
SBV (Speicherbereichsverteilung) 17
Schaltfunktion 34
Schickard 1, 5
Schieberegister 131, 170, 181
Schleifenanweisungen 113
Schlüsselwort 109
Schnittstelle 51, 74, 79 ff.
Schottky-TTL 177
Schreib-Lese-Speicher (RAM) 17, 37, 38, 70, 123, 166, 167 f.
Schreibregeln (BASIC) 108
Schreibsignal 147
Schreibverfahren 158 f.
Schrittgeschwindigkeit 127
Schrittweite 114
Schulpraxis (μC) 48 ff.
Schulpraxis (TR) 32 ff.
Schwierigkeitsindizes 8, 12
Scientific Computer 59
Scientific Notation 36
Sedezimaläquivalent 72
Sektoren 163
Sensortasten 35
Sequentieller Ablauf 57, 140
Sequenzer 139
Serielle Schnittstelle 79, 127
Serielle Speicher 166
Serielle Verarbeitung 59
Serienport 142
Shift 142
Sicherungszeichen 160 f.
Sicherung von Daten 160 ff.
Sieben-Segment-Anzeige 71
Signalpegel 132
Simulation 34, 137
Single Board Computer (SBC) 136
Single Step 135
Slice 138
Slice-Prozessor 138
Slowstep 135
Software 6, 31, 49, 55 f., 60
Software-Interrupt 99, 101
Solid State Software 64
Sonderzeichen 109
SOS-Technologie 179
Source 139, 173
Source Code 76
Source Program 102
Space 81
Spaltenformate 115 f.
Spannungsschnittstelle 79 ff.
Speicher (μC) 55 ff.
Speicher (TR) 37
Speicherbereichsverteilung (SBV) 17
Speicherdichte 157 f.
Speicherkontrolleinheit 139
Speicherkosten 182
Speicherlöschtaste 37
Speicherresident 56
Speicherverteilung 17
Spezifischer Widerstand 16
Splitting-Verfahren 26
Sprachelemente (BASIC) 109
Sprunganweisungen 112
Sprungbefehle 70, 140
SSI (Small Scale Integration) 63
Stack 29, 46, 50, 70, 71, 87, 97, 124
Stack-Adressen 87
Stack-Lift 29
Stackpointer 70, 71, 87, 97
Stackregiter 28 f., 46
Standardabweichung 8, 10
Standard-E/A 93
Standardfunktionen 111
Standard-Spaltenformat 115
Standard-TTL 178
Stapelbereich 71, 87, 124
Stapelspeicher (Stack) 50
Stapelübertragung 160
Stapelzeiger 87, 124
Startbit 156
Statement 140
Statische Speicher 166, 167 f.
Status 147
Statusbit 70, 147
Statusregister 70
Steckmodule 64
Steueranweisungen 112
Steuerbus 70 f., 87, 121
Steuerprogression 25
Steuersignale 56 f.
Steuerunterprogramm 149
Steuerwerk 44, 45, 55 f., 70 f., 121
Steuerzeichen 160
STOP-Anweisung 114
Stopbit 156
Store 30, 37
String 73
String-Manipulation 73
Stringvariablen 110
Strobe-Signal 92
Stromschleife 74, 79
Stromschnittstelle 79 ff., 127, 134
Stromschritt 127, 156
Struktogramm 49, 104 f.
Strukturierte Programmierung 64, 104 f.
Strukturierung 104
Subjektive Fehler 41
Subroutine 149
Symbolische Adresse 127
Synchrone Stapelübertragung 160 f.
Synchronisieren 78
Synchronisierung 87
Syntaktischer Fehler 73
Systembus 82 f.
System-Clock 69
Systemkomponenten (μC) 136
Systemtakt 69, 124
Systemterminal 75
S-100 Bus 82 f.

- TAB-Anweisung 116 f.
Tabelle 110
Tabellenformat 116
Tabulatorfunktion 116
Takt 69, 146
Takteinrichtung 69
Takten 78
Taktfrequenz 69, 124
Taktverarbeitung 124
Tandem 135
Taschen-Computer 2
Taschenrechner-Logiken 44 f.
Taschenrechner mit AOS 44 ff.
Tastatur (μC) 68
Tastatur (TR) 35
Tastendecodierer 44, 45
Tastenprellen 35
Taylor-Reihe 15
Technisch-wissenschaftliche DV 58
Teilnehmerbetrieb 95
Telegrafiergeschwindigkeit 127
Teletype (TTY) 79, 135
Terminal 56, 136
Term-Schreibweise 36
Text-Editor 73
Thomson-Brücke 16
Three-State-Schaltung 94, 180
Time Out Interrupt 96
Timer 131
Time Sharing 95
TK80-Schreibverfahren 156
Tonfrequenzbereich 155
Trace 77
Tracing 74
Transfer 57
Transferbefehle 30
Transistor 61
Transistor-Transistor-Logik (TTL) 62, 134, 177 f.
T-Register 29
Triggern 78
TR im Unterricht 32 ff.
Tristate 94, 180
Trommelspeicher 182
TRS 80-Schreibverfahren 157
True 140
TTL (Transistor-Transistor-Logik) 62, 134, 177 f.
TTY 135
TTY-Schnittstelle 79, 127, 134
Two Frequency Recording 159

- Übersetzer-Programm 150
Übersetzung 73
Übertragungsprozedur 160 f.
Übertragungsrate 156
Übertragung von Daten 160 ff.
Umgekehrte Polnische Notation (UPN) 28 ff., 36, 44, 46, 50
Umkehrfunktion 41
Umschalttaste 35
Underflow 39
Unformatiert 158
Unipolartechnik 166
Universalcomputer 67
Unterprogramm 49, 149
Unterprogrammaufruf 96
Unterprogrammebenen 17
Unterricht mit TR 32 ff.
Unwahr 140
UPN 28 ff., 36, 44, 46, 50
Ursprung 139

- Variablen 110
Variables Spaltenformat 116
Vektor 98, 110
Vergleichsoperator 113
Vergrabener Kollektor 178
Verkettung 40
Verschachtelung von Interrupts 100 f.
Verträglichkeit 134
Verzweigung 140
VIA 130 ff.
Videointerface 136
Vierspezies-Rechner 5, 6
VLSI (Very Large Scale Integration) 63
Vorwärtssprung 112
V.24-Schnittstelle 74, 79 ff., 127, 134
V.24-Steckerbelegung 80
V.28 79, 81

- Wahr 140
Wahrheitstabelle 172
Wechselphasencodierung 132, 133
Wechselschrift 159
Wechseltaktschrift 159
Wennanweisung 113
Wheatstone-Brücke 16
Widerstandsthermometer 15 f.
Winkel-Funktionen 38
Winkelmodus 38
Wissenschaftliche Notierung 36
Wortbreite 71, 122
Wortlänge 138
Wortorganisation 138
WRITE 147

- X-Register 29, 45 f.

- Y-Register 29, 45 f.

- Zahldarstellung 41
Zahlensystem 1, 21 ff.
Zehnerexponentialschreibweise 109 f.
Zehner-Logarithmus 38
Zeichenkette 73
Zeichenvorrat (BASIC) 109
Zeiger 98
Zeilennummer 108
Zeilenvorschub 117
Zeitgeber 131
Zeit-Interrupt 95
Zeit-Multiplex 93
Zeitscheiben-Technik 95
Zelle 172
Zentraleinheit 50, 55 f.
Ziel 139
Zinseszinsrechnung 118
Z-Register 29
Zugriffszeit 140, 182
Zuordnungsanweisung 112
Zuse 1, 5
Zweiersystem 21
Zweig 139
Zyklische Blocksicherung 165
Zyklus 69
20 mA Current Loop 84
20 mA-Schnittstelle 74, 79 ff., 127
20 mA-Stromschleife 134

VIEWEG

## Reihe Informationstechnik

Herausgegeben von Harald Schumny

Harald Schumny

**Signalübertragung**

Lehrbuch der Nachrichtentechnik und Datenfernverarbeitung. Mit 496 Abb. 1978. XVI, 404 S. DIN C 5 (Viewegs Fachbücher der Technik). Kart. 48,– DM.
Das Lehrbuch „Signalübertragung" ist vor dem aufgezeigten Hintergrund Telekommunikation entstanden, weshalb neben der ‚klassischen Nachrichtentechnik' (analoge Übertragungstechnik) auch digitale Übertragungstechniken wie Pulscodemodulation (PCM) und Datenfernübertragung besprochen werden.

**Arbeitsbuch Digitale Datenverarbeitung**

Unter Mitarbeit von Hagen Fiene und Karl-Heinz Wilke. Mit 67 Arbeitsblättern. 1976. 88 S. DIN C 5 (Viewegs Fachbücher der Technik). Kart. 12,80 DM.

**Digitale Datenverarbeitung für das technische Studium**

Mit 350 Abb. 1975. XVI, 352 S. DIN C 5 (Viewegs Fachbücher der Technik). Kart. 32,– DM.

Wolfgang Schneider

**BASIC**

Einführung für Techniker. 2., durchges. Aufl. 1979. VI, 134 S. DIN C 5 (Viewegs Fachbücher der Technik). Kart. 19,80 DM.
Das Buch vermittelt elementare Grundkenntnisse in der Programmiersprache BASIC, die der Schüler, Student, Techniker oder Ingenieur in der Ausbildung benötigt, um einfache mathematisch-naturwissenschaftliche Probleme selbständig programmieren zu können.

**FORTRAN**

Einführung für Techniker. 2., durchges. Aufl. 1979. IV, 129 S. DIN C 5 (Viewegs Fachbücher der Technik). Kart. 19,80. DM.

**Ausführliche Informationen erhalten Sie von Ihrem Buchhändler oder benutzen Sie die am Schluß des Buches eingeheftete Bestellkarte.**

VIEWEG

## Handbuch der Computer-Anwendung

Auswahl und Einsatz der EDV im Klein- und Mittelbetrieb

Von Erwin Grochla, Hans Gürth, Helmut Lehmann, Manfred Puhlmann, Michael Reicherts, Ernst Tiemeyer, Helmut Weber.

1979. 1178 S. DIN A 4. Im Arbeitsordner.
**Subskriptionspreis bis 31. Dezember 1979 DM 176,–** danach DM 220,–

Dieses Handbuch ist auf die praktische Problemsituation kleiner und mittlerer Betriebe zugeschnitten. Es wendet sich an Führungskräfte in diesen Betrieben und ist so aufgebaut, daß es als Arbeitsbuch für den Praktiker, als Lernprogramm zum Selbststudium und als Kompendium zum Nachschlagen benutzt werden kann. Dabei ist es für die Praktiker, die noch keinerlei Erfahrung mit EDV besitzen, ebenso konzipiert wie für diejenigen, die über Grundkenntnisse verfügen und vielleicht sogar schon eine EDV-Anlage einsetzen.
Für den Klein- und Mittelbetrieb ist der Einsatz von EDV in den letzten Jahren realisierbar geworden, ohne daß er gleichzeitig die Einstellung von EDV-Spezialisten erforderlich macht. Das „Handbuch der Computer-Anwendung" gibt dem Interessierten die notwendige Hilfestellung, die konkreten Probleme von der Auswahl des richtigen Computers bis zum täglichen Einsatz in den Griff zu bekommen.

**Ausführliche Informationen erhalten Sie von Ihrem Buchhändler oder benutzen Sie die Bestellkarte am Schluß dieses Buches.**

VIEWEG

Aus dem Programm:

## Mathematik – Didaktik

Johan van Dormolen
**Didaktik der Mathematik**
Mit einem Geleitwort von Erich Wittmann. (Didaktek van de wiskunde, dt., aus dem Niederl. übers. von Sigmar Kempfle.) 1978. X, 198 S. 108 Abb. Kart. 24,80 DM.

Gerhard Müller und Erich Wittmann
**Der Mathematikunterricht in der Primarstufe**
Ziele Inhalte, Prinzipien, Beispiele. 2., durchges. Aufl. 1978. XII, 320 S. 295 Abb. Kart. 29,80 DM.

Erich Wittmann
**Grundfragen des Mathematikunterrichts**
5., neu bearb. Aufl. 1978. VII, 202 S. Kart. 22,80 DM.
Dieses einführende Buch wendet sich an Mathematiklehrer und Lehrerstudenten aller Stufen sowie an Mathematiker, Psychologen und Pädagogen, die sich mit der Weiterentwicklung des Mathematikunterrichts befassen bzw. an ihr interessiert sind.

Udo Bromm
**Programmierbare Taschenrechner in Schule und Ausbildung**
Grundlagen und Anwendungen des Programmierens. Mit über 50 Programmen. 1979. VIII, 198 S. Kart. 26,80 DM.
Das Buch erschließt die Möglichkeiten des PTR für den Leser, der mathematische Kenntnisse der Sekundarstufe I mitbringt und den Rechner noch nicht unter spezifisch beruflichen Gesichtspunkten benutzt. Das Buch ist unabhängig von bestimmten Rechnertypen.

Alexander und Ursula Wynands
**Elektronische Taschenrechner in der Schule**
Ein Arbeits- und Aufgabenbuch für Lehrer und Schüler. 1978. VI, 125 S. Kart. 19,80 DM.

**Fragen Sie Ihren Buchhändler nach diesen Büchern oder benutzen Sie die am Schluß eingeheftete Bestellkarte.**

## Nicht sprachlos im Urlaub

Urlauber, die ins Ausland fahren, haben häufig keine Sprachkenntnisse oder nur noch eine sehr schwache Erinnerung an eine Fremdsprache, die sie einmal gelernt haben. Sie wollen oder müssen sich unter anders sprechenden Menschen bewegen, ohne dabei „sprachlos" zu sein. Für diesen Kreis bietet das ILS/Institut für Lernsysteme im Verlag Vieweg Urlaubs-Sprachkurse an, die entwickelt worden sind, um stets die wichtigsten Redewendungen im Kopf und zur Hand zu haben. Alle Redewendungen sind in szenische Dialoge eingebettet, damit man Sprechen, Hören und Verstehen trainieren kann. Jeder Kurs besteht aus einer Kassette mit Dialoglektionen und Übungen, sowie einem Textbuch mit 6seitigem Lernleitfaden, den Dialoglektionen und Übungen, einem Vokabelverzeichnis und einer Auflistung von Redeabsichten und Dialogverweisen. Die Kurse sind z.Z. für die Sprachen Englisch, Französisch und Spanisch lieferbar. Italienisch ist in Vorbereitung.

**Englisch · Französisch · Spanisch**
**je DM 23,80**

**Fragen Sie Ihren Buchhändler nach diesen Kursen oder benutzen Sie die am Schluß eingeheftete Bestellkarte.**

# Die individuelle Lösung

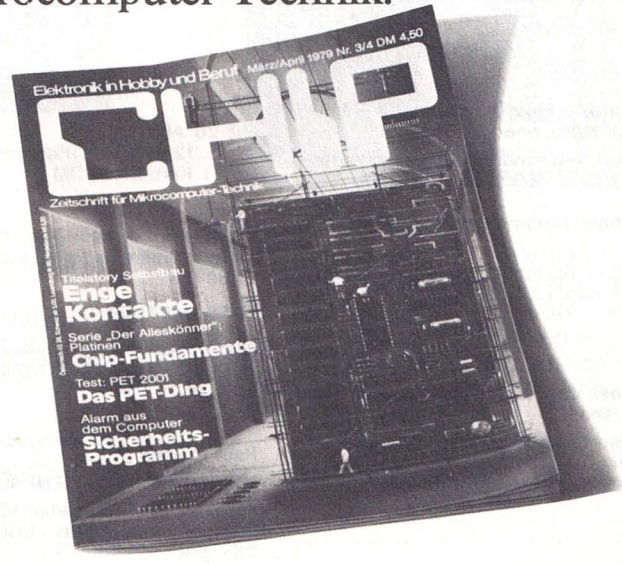
**... mit Standard-Komponenten oder nach Maß.**
**MCS liefert Hard- und Software aus einer Hand.**
**MCS berät, entwickelt und produziert.**
**MCS führt Seminare und Schulungen durch.**

MCS hat das Know-how in der Entwicklung und Herstellung von Microcomputer-Systemen. MCS-Microcomputer ALPHA 1 und BETA 8: Ausdruck beständiger Leistung.

**MCS**

Micronic Computer Systeme GmbH, Zentralflughafen Tempelhof
Tempelhofer Damm / Einfahrt Paradestraße · 1000 Berlin 42

CHIP, die Zeitschrift für Mikrocomputer-Technik. Für alle, die mehr wissen wollen und mehr wissen müssen, wenn es um Mikrocomputer geht. In Hobby und Beruf. Denn CHIP bringt umfassend und verständlich alles, worauf es ankommt: Hintergrundberichte, Beispiele, umfassende Anleitungen, Software, Nachrichten, Trends und Meinungen. Ausgabe für Ausgabe eine geballte Ladung an Wissenswertem und Wichtigem. Damit Sie Bescheid wissen und mitreden können. CHIP, die Zeitschrift für Mikrocomputer-Technik.




CHIP gibt es monatlich im Zeitschriften- und Buchhandel oder einfach bestellen beim CHIP-Leserservice, Vogel-Verlag, Postfach 67 40, 8700 Würzburg 1

VIEWEG

# Taschenrechner-Literatur

## Anwendung programmierbarer Taschenrechner

Die Reihe „Anwendung programmierbarer Taschenrechner" bietet den Benutzern dieser Rechner eine breite Palette von Aufgabenstellungen aus den den Anwendungsgebieten der Natur- und Wirtschaftswissenschaften an, für die Rechnerprogramme zur numerischen Lösung dargestellt werden.

Helmut Alt

**Angewandte Mathematik – Finanzmathematik – Informatik – Statistik für UPN-Rechner**

1979. VIII, 158 S. DIN C 5 (Anwendung programmierbarer Taschenrechner, Bd. 1). Kart. 29,80 DM

**Allgemeine Elekrotechnik, Nachrichtentechnik für UPN-Rechner**

1979. Ca. 120 S. DIN C 5 (Anwendung programmierbarer Taschenrechner, Bd. 2). Kart. ca. 25,– DM

Peter Kahlig

**Mathematische Routinen der Physik, Chemie und Technik für AOS-Rechner, Teil I**

Mit universellen Sonderprogrammen zum Zeichnen und Drucken sowie 71 Abb. 1979. VI, 178 S. DIN C 5 (Anwendung programmierbarer Taschenrechner, Bd. 3/I). Kart. 38,– DM

---

Konrad Hoyer und Gerhard Schnell

**Differentialgleichungen der Elektrotechnik**

Lösung mittels Theorie der Differentialgleichungen, Laplace-Transformation und programmierbarer Taschenrechner. Mit ca. 60 Abb. 1978. VIII, 121 S. DIN C 5. Kart. 16,80 DM

## Programmieren von Taschenrechnern

Mit diesen Büchern werden dem im programmieren unerfahrenen Leser Kenntnisse über den Umgang mit programmierbaren Taschenrechnern vermittelt. Jeder Band ist auf bestimmte Rechnertypen oder Familien zugeschnitten.

Hans Heinrich Gloistehn

**SR 56**

Mit 26 Abb. u. zahlr. Tab. 2., durchges. Aufl. 1978. IV, 140 S. 12 x 19,5 cm.(Programmieren von Taschenrechnern, Bd. 1). Kart. 21,80 DM

**TI-57**

Mit 25 Abb. u. zahlr. Tab. 1978. IV, 112 S. 12 x 19,5 cm. (Programmieren von Taschenrechnern, Bd. 2). Kart. 21,80 DM

**TI-58 und TI-59**

Mit 25 Abb. u. zahlr. Tab. 2., verb. Aufl. 1979. IV, 150 S. 12 x 19,5 cm. (Programmieren von Taschenrechnern, Bd. 3). Kart. 24,80 DM

Paul Thießen

**HP-29C/HP-19C und HP-67/HP-97**

Hrsg. Hans H. Gloistehn. 1979. Ca. 160 S. 12 x 19,5 cm. (Programmieren von Taschenrechnern, Bd. 4). Kart. 25,– DM

Hans-Joachim Ludwig

**Programmoptimierung für AOS-Taschenrechner**

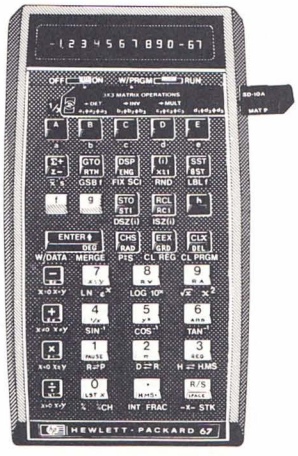
1979. VIII, 96 S. 12 x 19,5 cm (Programmieren von Taschenrechnern, Bd. 5). Kart. ca. 19,80 DM

**Bitte benutzen Sie für Ihren Auftrag die nebenstehende Bestellkarte.**

# Die Programmierbaren von Hewlett-Packard

# HP-67 HP-97

## Mit intelligentem Kartenleser . . . und mit Drucker



- 224 Programmzeilen, bis zu 3 Programmschritte pro Zeile
- 26 adressierbare Datenregister
- Speichert und liest Programme und Daten (Magnetkarten zwischen HP-67 und HP-97 voll austauschbar)
- Umfassende Korrekturmöglichkeiten
- Netz- und Batteriebetrieb (bis 7 Stunden)

Aber HEWLETT-PACKARD bietet noch mehr. Ergänzend zu den Rechnern liegt ein umfassendes Software-Angebot von mehr als 3000 Programmen aus Wirtschaft, Wissenschaft und Technik in praxisgerechter Form vor. Damit eröffnet sich den Benutzern von HEWLETT-PACKARD-Rechnern die Möglichkeit, die individuellsten Problemstellungen sicher und schnell zu lösen.

HEWLETT hp PACKARD

Hewlett-Packard GmbH/Vertriebszentrale, Berner Str. 117, 6000 Frankfurt/M. 56, Tel. (0611) 50 04-1

**Plant Ihre Termine**
**Verfolgt Ihre Kosten**
**Analysiert Ihre Daten**
**Lernt mit Ihnen Englisch**

**. . . und spielt mit Ihnen Schach**

Das Homecomputer System TI-99/4 eröffnet zum ersten Male dem individuellen Anwender den Zugang zu interaktiver Computerleistung ohne DV-Kenntnisse. Es erreicht dies durch weitgehenden Einsatz von Solid State Software (Anwenderprogrammen in Halbleiterspeichern).

Seine Zentraleinheit enthält:
- die Programmiersprache TI-Basic mit einer Reihe graphischer Befehle zur Steuerung des Farbbildschirmes
- einen interaktiven Formelrechner
- einen programmierbaren Tongenerator
- die Prozessorbausteine der TI 990-Familie

Seine einsteckbaren Command Module (6—30 kByte)
- ersparen alle Ladevorgänge, machen den TI-99/4 unmittelbar betriebsbereit
- steuern den Dialog mit dem Anwender ohne Fehlermöglichkeit über Text, Farbbild und Ton (Sprachmodule)

Sein Arbeitsspeicher von 16K RAM
- enthält dadurch keine Programme (außer denen, die Sie selber schreiben und auf Kassette speichern wollen)
- steht damit vollständig für Ihre Daten zur Verfügung.

Vorführung und Information in den Fachabteilungen der Warenhäuser und im Fachhandel oder direkt von

TEXAS INSTRUMENTS

Deutschland GmbH
Haggertystraße 1 · 8050 Freising


ISBN 3-528-04154-4